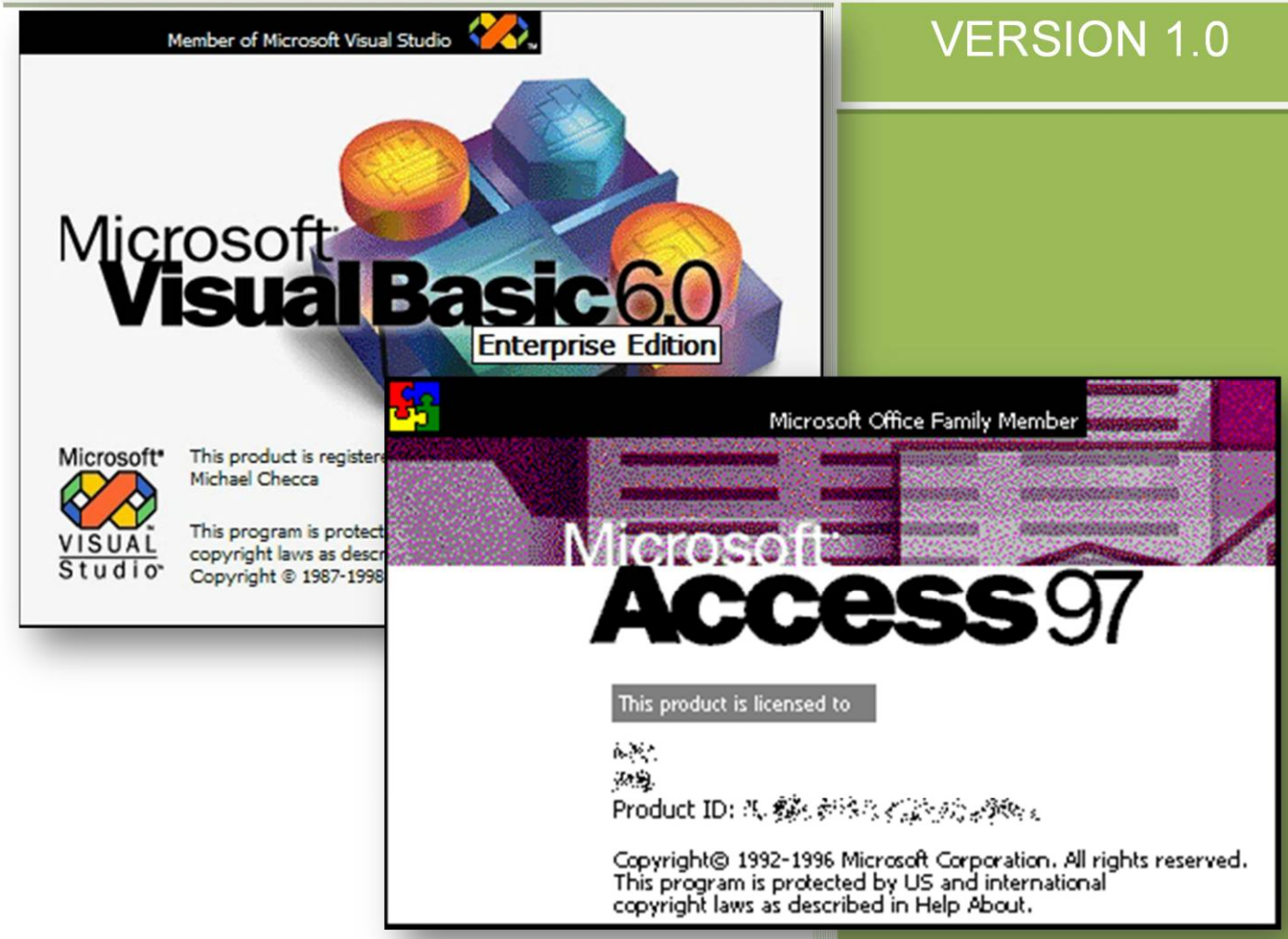
2011

# MANUAL BASICO DE VISUAL BASIC 6.0 Y DISEÑO DE BASE DE DATOS EN ACCES

VERSION 1.0




**DAVID ENRIQUE CELIS BERNAL**

**ANALISTA Y PROGRAMADOR DE SISTEMAS**

## TABLA DE CONTENIDO


**PROLOGO** 4

**1. VISUAL BASIC 6.0** 5

**2. TIPOS DE VARIABLES** 17

**3.INICIO DE VISUAL** 18

**4. PROPIEDADES DEL FORMULARIO** 19

**5. LABEL O ETIQUETA** 19

**6.MI PRIMERA ANIMACION** 20

**7. GUARDADO DE PROYECTOS** 23

**8. IMAGE** 24

**9. TIMER** 25

**10. COMMAND BUTTON** 27

**11. TIPOS DE VARIABLE 2** 29

**12. TIPO DE VARIABLES 3** 29

**13. TEXTBOX CAJAS DE TEXTO** 30

**14. TIPOS DE INTERFAZ** 32

**15. CON EL RIGHT CLICK IMAGE CONVERTER** 41

*16. CREACION DE MENUS* 47

**17. TEXTO CON MARQUESINA** 51

**18. ANIMACION CON REBOTE DE BALON** 53

**19. EJEMPLO DE EJERCICIO DE UN MINI PROCESADOR DE TEXTO** 54

**20. CHECKBOX Y OPTION BUTTON** 57

**21. SENCILLO PROGRAMA DE COLORES Y POSICIONES** 58

**22.COLORES RGB** 60

**23. USO DE OCX ALTERNAS DE VISUAL BASIC** 64

**24. INSTRUCCIONES DE REPETICION** 70

**25. BASE DE DATOS ACCES** 72

**26. CONTROL DE HERRAMIENTAS DAO** 92

**27. BOTON DE BORRADO DE REGISTRO UNO A UNO** 107

**28. BORRADO GENERAL DE DATOS, MACROS E IMPRESIÓN DE INFORMES** 120

**SEGUNDO EJEMPLO DE ACUMULADOS Y MACROS EN ACCES** 134

**29. SPLASH O PANTALLA DE BIENVENIDA** 142

**30. INCIO DE SECION AL SOFTWARE** 144

**31. CREACION DE AUTO ASISTENTES** 150

**32. CREACION DEL EJECUTABLE** 156

**33. INSTALADOR** 157

**34. MANEJO DE INTERNET, WORD, EXCEL DESDE VISUAL** 164

**35. OBJETO OLE** 167

**36. VIDEOS A VISUAL BASIC** 170

**37. CARGAR LA HORA Y LA FECHA A UNA CAJA DE TEXTO** 174

**38. CARGAR CONTROL CALENDARIO** 175

**39. INFORMES DESDE VISUAL UTILIZANDO DATA ENVIRONMENT Y DATA REPORT** 177

**40. ABRIR Y CERRAR UNIDAD DE CD DESDE UN BOTON** 187

**41. APLICACIONES CLIENTE SERVIDOR** 189

# PROLOGO

**Debido al avance arrollador de la tecnología las personas han querido actualizarse en sus funciones diarias queriendo aplicaciones que faciliten su trabajo y lo puedan cumplir de una manera eficiente y rápida sin demoras, también poder almacenar electrónicamente sus datos para hacerlos más fácil de utilizar, con el fin de llegar a este objetivo aparecen los analistas y desarrolladores de sistemas de información que crean dichas aplicaciones a medida de las necesidades del usuario.**

**Esta es la primera versión de este mini libro de programación orientada a objetos en visual Basic 6.0, y además diseño de base de datos en el motor Access, cuenta con gráficas, explicación detallada de controles, ejemplos de mini aplicaciones con absolutamente todo el código fuente de la misma.**

**Esta guía le servirá al programador como manual complementario a las funciones que diario realiza, como muchas veces quedan dudas también en internet se puede consultar en la web del programador, foros donde también nos orienta a la elaboración de nuestras aplicaciones.**

# 1. VISUAL BASIC 6.0

**Visual Basic es uno de los tantos lenguajes de programación que podemos encontrar hoy en día. Dicho lenguaje nace del Basic (beginner´sall-purpose symbolic instruction code) que fue creado en su versión original en el dartmouthcollege, con el propósito de servir a aquellas personas que estaban interesadas en iniciarse en algún lenguaje de programación. Luego de sufrir varias modificaciones, en el año 1978 se estableció el Basic estándar. La sencillez del lenguaje ganó el desprecio de los programadores avanzados por considerarlo "un lenguaje para principiantes".**

**Primero fue gw-basic, luego se transformó en QuickBasic y actualmente se lo conoce como visual Basic y la versión más reciente es la 6 que se incluye en el paquete visual estudio 6 de Microsoft. Esta versión combina la sencillez del Basic con un poderoso lenguaje de programación visual que juntos permiten desarrollar robustos programas de 32 bits para Windows. Esta fusión de sencillez y la estética permitió ampliar mucho más el monopolio de Microsoft, ya que el lenguaje sólo es compatible con Windows, un sistema operativo de la misma empresa.**

**Visual Basic ya no es más "un lenguaje para principiantes" sino que es una perfecta alternativa para los programadores de cualquier nivel que deseen desarrollar aplicaciones compatibles con Windows.**

**En este informe explicaremos algunos términos y/o características de mismo con la finalidad de aprender más sobre este programa y manejarlo con facilidad**

1. **De los fundamentos de programación explique:**

1. **Es un lenguaje de programación que se ha diseñado para facilitar el desarrollo de aplicaciones en un entorno grafico (gui-graphical user interface) como Windows 98,** Windows **o superior.**
2. **¿qué es visual Basic?**

   **Diseñador de entorno de datos: es posible generar, de manera automática, conectividad entre controles y datos mediante la acción de arrastrar y colocar sobre formularios o informes.**

   **Los objetos actives son una nueva tecnología de acceso a datos mediante la acción de arrastrar y colocar sobre formularios o informes.**

**Asistente para formularios: sirve para generar de manera automática formularios que administran registros de tablas**

**O consultas pertenecientes a una base de datos, hoja de cálculo u objeto (ado-active data object)**

**Asistente para barras de herramientas es factible incluir barras de herramientas es factible incluir barra de herramientas**

**Personalizada, donde el usuario selecciona los botones que desea visualizar durante la ejecución.**

**En las aplicaciones html: se combinan instrucciones de visual Basic con código html para controlar los eventos que se realizan con frecuencia en una página web.**

**La ventana de vista de datos proporciona acceso a la estructura de una base de datos. Desde esta también acceso al diseñador de consultas y diseñador de base de datos para administrar y registros.**

3. **Características de visual Basic.**

- **Barra de título: muestra el nombre del proyecto y del formulario q se está diseñando actualmente**
- **Barra de menús: agrupa los menús despegables que contienes todas las operaciones que pueden llevarse a cabo con visual Basic 6.0.**
- **Barra de herramientas estándar: contienen los botones que se utilizan con mayor frecuencia cuando se trabaja con un proyecto. Simplifica la elección de opciones de los menús archivo, edición, ver y ejecutar; además, en el área derecha presenta la ubicación (coordenadas) y el tamaño del objeto seleccionado**
- **Ventana de formulario: es el área donde se diseña la interfaz gráfica, es decir, es donde se inserta electo gráficos, como**
- **Botones,** imágenes, casilla **de verificación, cuadros de listas, etc.**
- **Cuadro de herramientas: presenta todos los controles necesarios para diseñar una aplicación, como cuadros de texto, etiquetas, cuadros de listas, botones de comandos, etc.**
- **Ventana de proyecto: muestra los elementos involucrados en el proyecto, como formularios, módulos, controles ocx, etc. Cada elemento puede seleccionarse en forma independiente para su edición.**
- **Ventana de posición del formulario: muestra la ubicación que tendrá el formulario en la pantalla, cuando ejecute la aplicación. Esta ubicación puede cambiarse si se hace clic con el botón izquierdo del mouse.**

**La ventana propiedades muestra todas las propiedades**

**Del control actualmente seleccionado, en este caso muestra las propiedades del form1, luego podemos ver que abajo dice "form1 form", lo que está en negrita es el nombre del objeto, y lo que le sigue es el tipo de objeto, en este caso es un formulario (form)**

4. **Mencione y explique las partes del entorno de visual Basic.**
5. **Mencione y explique los botones de la barra de herramientas de visual Basic.**

Cursor del ratón: sirve para quitar la selección del que hayamos pulsado por error.

**Picture:** permite incluir gráficos, es parecido al image pero con variantes.

Label (Etiqueta): sirve para poder poner mensajes o textos en la pantalla, pero no son editables por el usuario.

**Textbox:** caja de texto, permite que el usuario pueda escribir directamente.

**FRAME:** es un contenedor para los demás controles.

Commandbutton: botón de comando, permite que el usuario pueda pulsar botones y que realicen determinadas acciones

**Checkbox:** caja de chequeo, permite tener opciones sobre las que el usuario confirma o niega.

**Optionbutton:** botón de opción, permite tener varias opciones de las cuales seleccionar una.

**Combobox:** caja combinada, permite tener varias opciones de las cuales seleccionar y ver una.

**Listbox:** caja de lista, presenta una lista de opciones y permite seleccionar una o varias De dichas opciones

**Hscroll:** (barra horizontal de scroll) se puede crear una barra de desplazamiento horizontal.

**Vscroll:** barra vertical de scroll, permite tener una barra vertical para desplazamientos.

**Timer:** es un controlador de tiempo.

**Drivebox:** caja de unidades, permite indicar que unidades de almacenamiento tiene el ordenador.

**Dirbox:** caja de directorios, indica los directorios y subdirectorios de una ruta en concreto.

**Filelist:** indica una lista de los ficheros que se encuentran en un determinado directorio o subdirectorio.

**Shape:** permite incluir formas gráficas en el formulario, como un cuadrado, un círculo, etc.

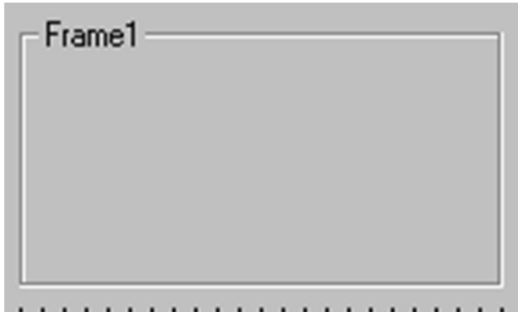
**Line:** permite incluir una línea en el formulario.

**Image:** permite incluir imágenes como fotografías, dibujos, etc...

**Data:** permite la conexión con una base de datos y poder añadir, modificar, consultar, borrar, etc...

**Cuando actives este botón y mientras no desbloquees los controles utilizando la misma opción no se podrán mover ninguno de los controles del formulario activo. Sin embargo en si abres otro formulario que no tenga los controles bloqueados si se podrán mover. Si añades más controles a un formulario bloqueado estos quedan bloqueados automáticamente**

**Tiene la siguiente forma:**




**Un control frame proporciona un agrupamiento identificable para controles. También puede utilizar un frame para subdividir un formulario funcionalmente por ejemplo, para separar grupos de controles optionbutton.**

**Check button y option button (botones de elección y opción)**

**Se obtienen directamente de la caja de herramientas.**

**Dada la similitud de ambos controles, se comentan conjuntamente.**

**El control checkbox, o casilla de verificación, permite elegir una opción (activada / desactivada, true/false) que el usuario puede establecer o anular haciendo click. Una x en una casilla de verificación indica que está seleccionada, activada, o con valor true. Cada casilla de verificación es independiente de las demás que puedan existir en el formulario, pudiendo tomar cada una de ellas el valor true o false, a voluntad del operador.**

**Un control optionbutton muestra una opción que se puede activar o desactivar, pero con dependencia del estado de otros controles optionbutton que existan en el formulario.**

**Generalmente, los controles optionbutton se utilizan en un grupo de opciones para mostrar opciones de las cuales el usuario sólo puede seleccionar una. Los controles optionbutton se agrupan dibujándolos dentro de un contenedor como un control frame, un control picturebox o un formulario. Para agrupar controles optionbutton en un frame o picturebox, dibuje en primer lugar el frame o picturebox y, a continuación, dibuje dentro los controles optionbutton. Todos los controles optionbutton que están dentro del mismo contenedor actúan como un solo grupo, e independientes de los controles optionbutton de otros grupos distintos.**

**Aunque puede parecer que los controles optionbutton y checkbox funcionan de forma similar, hay una diferencia importante: cuando un usuario selecciona un optionbutton, los otros controles del mismo grupo optionbutton dejan de estas disponibles automáticamente. Por contraste, se puede seleccionar cualquier número de controles checkbox.**

**List box y combo box**

**Estos dos controles, debido a su similitud, se estudian conjuntamente.**

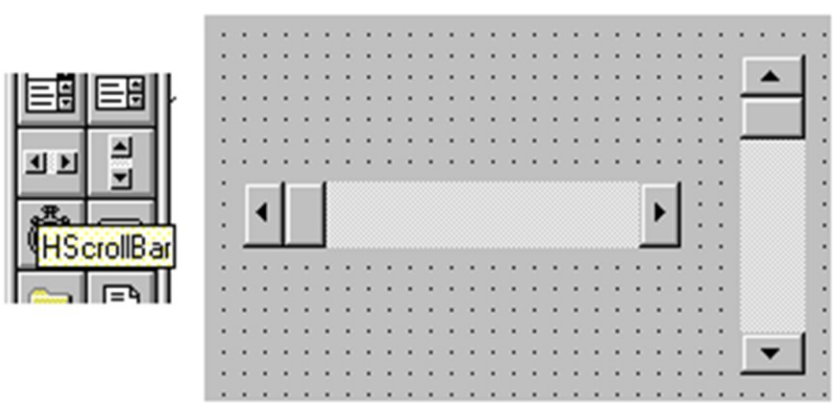
**Se obtienen directamente de la caja de herramientas:**

**Un control listbox muestra una lista de elementos en la que el usuario puede seleccionar uno o más. Si el número de elementos supera el número que puede mostrarse, se agregará automáticamente una barra de desplazamiento al control listbox.**

**Un control combobox combina las características de un control textbox y un control listbox. Los usuarios pueden introducir información en la parte del cuadro de texto y seleccionar un elemento en la parte de cuadro de lista del control. En resumen, un combobox es la combinación de un listbox, que se comporta como si de un listbox se tratase, y de un textbox, con comportamiento análogo a un textbox sencillo, con la particularidad aquí de que el texto se le puede introducir por teclado, o elegir uno de los que figuran en la parte listbox del combo.**

**Controles hscrollbar y vscrollbar**

**Son dos controles similares, para introducir un dato cuasi-analógico en una aplicación. Se toman directamente de la caja de herramientas, y tienen un aspecto parecido al de un control de volumen de un equipo de música. El hscrollbar está en posición horizontal, y el vscrollbar en posición vertical.**




**Mediante estos controles se pueden introducir datos variando la posición del cursor.**

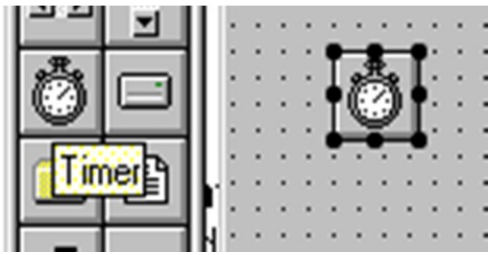
**Timer temporizador**

**Este objeto permite establecer temporizaciones. Presenta**

**Una novedad respecto a los controles estudiados hasta ahora. El control timer solamente se ve durante el tiempo de diseño. En tiempo de ejecución, el control permanece invisible.**

**La temporización producida por el timer es independiente de la velocidad de trabajo del ordenador. (Casi independiente. El timer no es un reloj exacto, pero se le parece)**

**Se toma directamente de la caja de herramientas, y tiene el aspecto siguiente:**




**Shape**

**Se toma directamente de la caja de herramientas:**

**Shape es un control gráfico que se muestra como un rectángulo, un cuadrado, una elipse, un círculo, un rectángulo redondeado o un cuadrado redondeado.**

**Utilice controles shape en tiempo de diseño en lugar o además de invocar los métodos circle y line en tiempo de ejecución. Puede dibujar un control shape en un contenedor, pero no puede actuar como contenedor. (esto quiere decir que un control shape nunca le servirá, por ejemplo, para albergar varios optionbutton y pretender que sean independientes de otros controles optionbutton que se encuentren fuera del control shape.**

**Este control no tiene procedimientos. En realidad, solamente sirve para mostrar un determinado gráfico, envolver gráficamente a otros controles, pero no tiene ninguna aplicación en cuanto a programa. Es un "adorno" para sus aplicaciones.**

**Line**

**Se toma directamente de la caja de herramientas**

**Line, al igual que shape, es un control gráfico que solamente sirve para poner una línea en un formulario. Del mismo modo, no tiene procedimientos, por lo que no sirve para aportar código al programa. Solo sirve para aportar una característica gráfica, es un adorno.**

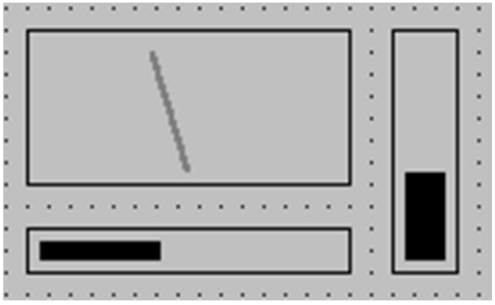
**Control gauge**

**Este control presenta una información numérica de forma gráfica, bien como un display lineal (típico por ejemplo en ecualizadores de audio), o como una aguja. No está normalmente en la caja de herramientas, por lo que hay que traerla desde los controles personalizados (menú desplegable de herramientas) se denomina microhelp gauge control. El archivo que lo contiene se denomina gauge16.ocx, 16 bits**

**Mediante este control, podemos presentar una magnitud numérica de una forma cuasi-analógica. Podríamos decir que es un control similar al hscrollbar, que en vez de meter información a la aplicación, la presenta.**

**Este control puede servir, por ejemplo, para presentar el tanto por ciento de ejecución de una tarea, como elemento tranquilizante. Puede presentar el nivel de un depósito de agua, etc.**

**Presenta las dos formas siguientes:**



**En la figura puede verse un gauge de aguja, uno de barra horizontal y otro de barra vertical. Para mejorar la presentación, el gauge permite poner un gráfico como fondo, cambiar el color de la barra, color de fondo, etc.**

**El control gauge crea medidores definidos por el usuario, que puede elegir entre los estilos lineales (relleno) o de aguja.**

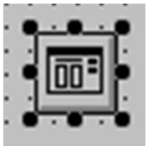
**Nota para la distribución cuando cree y distribuya aplicaciones con controles gauge, tendrá que instalar el archivo apropiado en el subdirectorio system de Windows del cliente. El kit para instalación que incluye visual Basic, le proporciona herramientas para escribir los programas que instalan las aplicaciones correctamente.**

**El commondialog es un control del que se libran muy pocas aplicaciones. Dada la importancia de este control, se le dedica un capítulo único en esta guía del estudiante.**

**Cuadro de dialogo commondialog**

**Normalmente se encuentra en la caja de herramientas**

**Este control no se presenta en tiempo de diseño más que con un simple icono:**



**El cuadro de diálogo, commondialog se utiliza para varias funciones:**

**Abrir ficheros**

**Guardar ficheros**

**Elegir colores**

**Seleccionar impresora**

- **Seleccionar fuentes**
- **Mostrar el fichero de ayuda**

**En realidad el cuadro de diálogo permite conocer datos con los cuales, y mediante el código adecuado, abriremos o guardaremos ficheros, elegiremos** colores **o seleccionaremos fuentes. Es decir, el commondialog no realiza más funciones que mostrar ficheros existentes, fuentes disponibles, colores, para que, mediante código, abramos esos ficheros o usemos una determinada fuente.**

**Dependiendo de la aplicación para la que vaya a usarse se deberá activar de distintas formas. Si el cuadro de diálogo se va a usar para seleccionar la** impresora **y para otras aplicaciones, es recomendable usar uno exclusivamente para seleccionar la impresora.**

**Esta última recomendación se debe a que, para el control de la impresora, el commondialog si realiza las funciones de selección de impresora predeterminada. Esta diferencia operativa hace que si usamos el mismo commondialog para seleccionar impresora y abrir ficheros, por ejemplo, se "cuelgue" el commondialog.**

5. **Defina los siguientes términos.**
6. **eventos: es una acción como hacer clic, doble clic, presionar una tecla, mover el puntero del mouse, etc. Que el usuario debe realizar para que un objeto ejecute una acción determinada cada control responde a diferentes eventos, algunos de ellos tienen características comunes. Los eventos pueden visualizarse en la ventana de código.**

   **Métodos: son procedimientos definidos en visual Basic para realizar operaciones específicas sobre los objetos (controles o formularios)**

   **Controles: son los objetos que conforman la interfaz gráfica de un programa;**

   **A través de ellos, un usuario interactúa con la aplicación. Sus características**

   **Pueden cambiarse por medio de la ventana propiedades**

   **Proyecto:**

   **Propiedades: son los datos que hacen referencia a un objeto o formulario. Ejemplo: color de fondo del formulario, fuente de texto de un textbox.**

   **Objetos: un objeto es una entidad que tiene asociado un conjunto de métodos, eventos y propiedades. Hay muchas clases de objetos, y por tanto, puede llegar a**

**haber tantos métodos, eventos y propiedades distintas como objetos diferentes. Ejemplo: una caja de texto (textbox) en la cual podemos escribir cualquier línea es un objeto.**

**Clases: una clase no es nada más que un objeto, este objeto, tiene propiedades, funciones y métodos. Para empezar ahora la creación de propiedades si se utiliza propertylet y propertyget; la diferencia es casi nada, inclusive podría decir que una clase en visual Basic, es casi lo mismo que un control, pero ahora nace una nueva pregunta, cuando utilizar un control y cuando utilizar una clase, bueno la opinión que voy a dar es desde mi perspectiva.**

**Módulo: un proyecto visual Basic no sólo está compuesto de formularios, sino también de lo que se denominan módulos.**
**Un módulo es un fichero visual Basic donde escribimos parte del código de nuestro programa, y digo parte, porque puede haber código en el formulario también.**

7. **De los fundamentos de programación explique:**

**Módulos de control:**

**Módulos de formularios:**

**Módulos de clases**

**Módulos estándar**

**Editor de códigos:**

**Variable: dim: al declarar una variable con esta palabra estamos diciendo que la variable sea local al ámbito en que se declara. Puede ser dentro de un procedimiento o dentro de un formulario, de esta forma no sería accesible desde los demás procedimientos o formularios.**

**Public: las variables declaradas serán públicas y podrán estar accesibles desde todos los formularios de la aplicación. Para conseguirlo tendremos que declararlas en un módulo de código, no en la sección declarations de cualquier formulario de los que conste la aplicación. Para crear un módulo de código en el menú principal de visual Basic marcamos en insert/module y aparecerá junto a los demás formularios de la ventana de proyecto aunque con un icono distinto indicando que se trata de un módulo de código.**

**Static: con esta forma de declarar variables conseguiremos que las variables locales no se creen y se destruyan al entrar y salir de los procedimientos donde fueron declaradas sino que se mantenga su valor durante todo el periodo de ejecución de la aplicación. De esta forma a entrar en algún procedimiento las variables recuerdan el valor que tenían cuando se salió de él.**

## 2. TIPOS DE VARIABLES

| Tipo | Comentario |
|---|---|
| Boolean | Sólo admite 2 valores true o false |
| Byte | Admite valores entre 0 y 255 |
| Integer | Admite valores entre -32768 y 32767 |
| Long | Admite valores entre -2.147.483.648 y 2.147.483.647 |
| Single | Admite valores decimales con precisión simple |
| Double | Admite valores decimales de doble precisión |
| Currency | Válido para valores de tipo moneda |
| String | Cadenas de caracteres |
| Date | Fechas, permite operar con ellas |

Constante: declaración de constantes que pueden ser usadas en cualquier punto en lugar de su valor, permitiendo cambiarlo cuando sea necesario, sin tener que cambiarlo en todos los sitios en que se utiliza. La expresión no puede utilizar llamadas a funciones, pues la constante se calcula en tiempo de compilación, no en tiempo de ejecución.

# 3.INICIO DE VISUAL

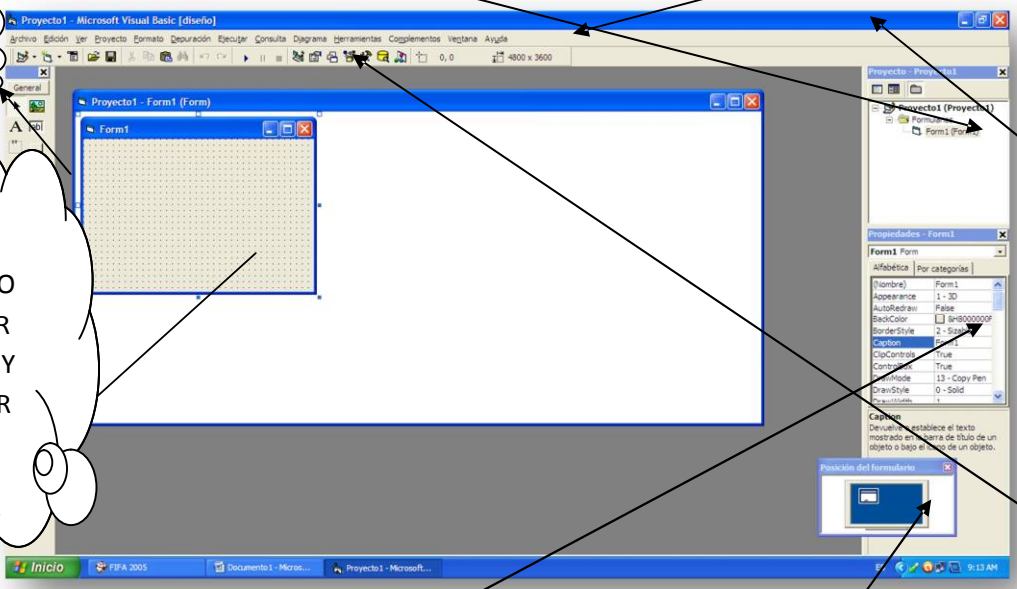
CLICK AQUÍ Y LUEGO EN EL BOTON ACEPTAR

CONTROLES PARA PROGRAMAR

BARRA VISTA DE FORMULARIOS

BARRA DE MENUS

ENTORNO DE DESARROLLO INTEGRADO (IDE)




BARRA DE TITULO

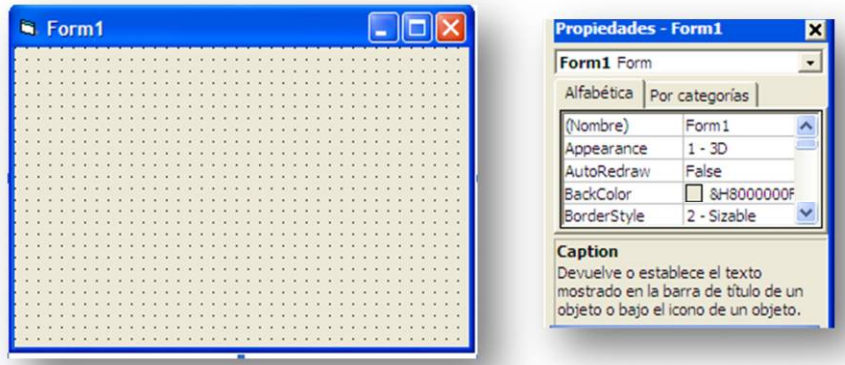
VENTANA FORMULARIO PARA SITUAR CONTROLES Y PROGRAMAR

VENTANA DE PROPIEDADES DE LOS CONTROLES

CUADRAR FORMULARIO EN TIEMPO DE EJECUCION

BARRRA DE HERRAMIENTAS

## 4. PROPIEDADES DEL FORMULARIO




**Caption:** nombre externo del formulario

**Name:** nombre interno que se usa para programar

**Backcolor:** color de fondo del formulario

**Borderstyle:** tipos de borde del formulario, para mostrar y ocultar los bordes

**Icons:** cambiar el icono por defecto del formulario

**Windowstate:** se utiliza para maximizar, inactivar, minimizar el formulario en tiempo de ejecución para que salga dicho formulario en pantalla completa al iniciar el programa se coloca la opción maximizar

**Picture:** imagen de fondo del formulario

## 5. LABEL O ETIQUETA

Propiedades

**Caption**

**Name**

**Bakcolor**

**Font**: tipo de letra del label

**Forecolor**: color de letra

**Backstyle:** estilo del label: más que todo se utiliza de tipo transparente

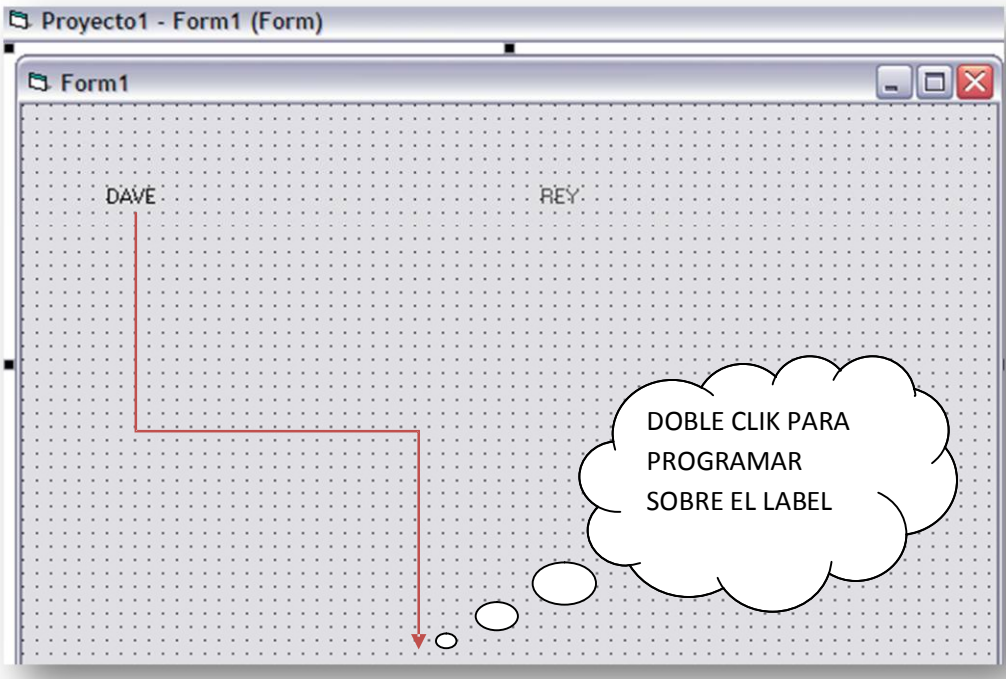
**Aligment:** para alinear el texto en ejecución

Ejemplo 1

## 6. MI PRIMERA ANIMACION

Nota: a la hora de programar a cada control es bueno colocarle un nombre para evitar inconvenientes en el momento de programar con su respectiva palabra por ejemplo: a una **caja de texto: txnombre**, a **un label: lbnombre** etc.

Se utilizaran 2 labels borderstyle tipo transparente y colocar en el caption de cada uno dos nombres en las propiedades

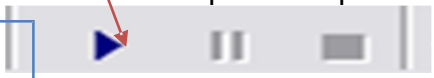


Proyecto1 - Form1 (Código)

Label1 | Click

```
Private Sub Form_Load()

End Sub

Private Sub Label1_Click()
Label2.ForeColor = &HFF0000
Label2.Caption = "SISTEMAS"
End Sub

Private Sub Label2_Click()
Label1.ForeColor = &HFF0000
Label1.Caption = "SISTEMAS"
End Sub
```

Para iniciar la ejecucion de la aplicación pulsamos la tecla de funcion f5 o en la barra de



*Herramientas* y esto es lo que hace

**Propiedades - Form1**

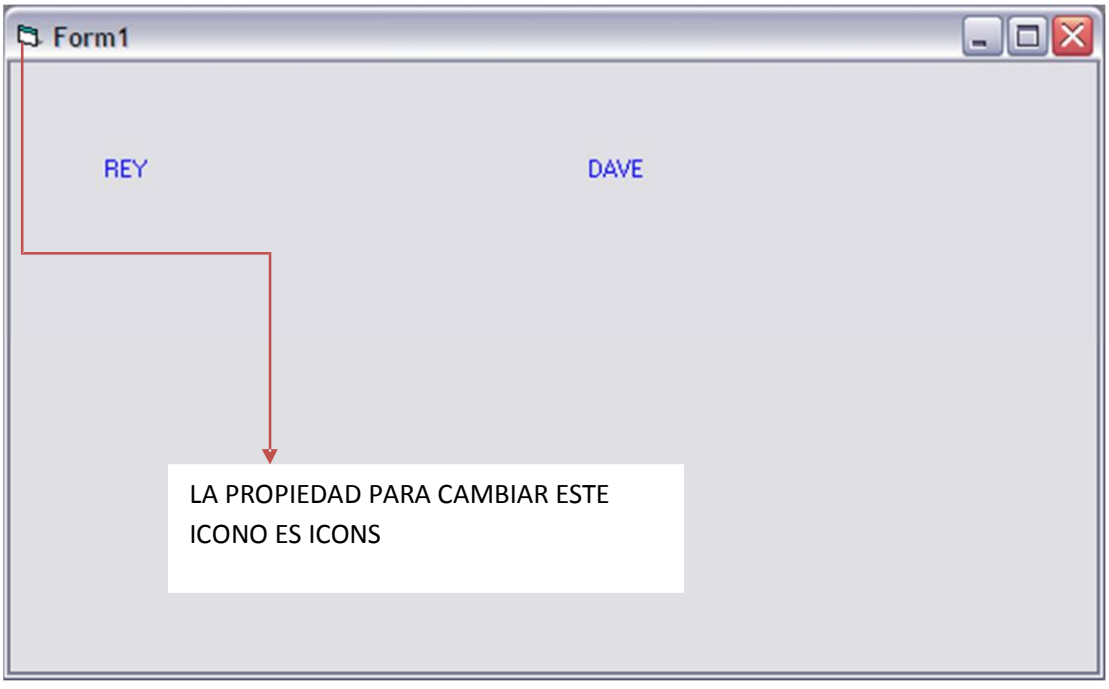
**Form1** Form

Alfabética | Por categorías

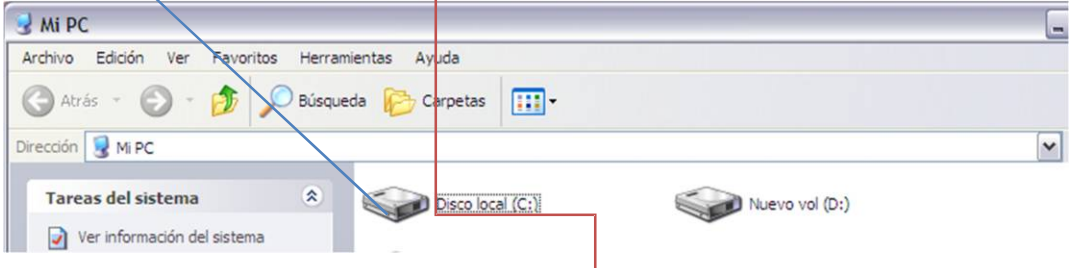
| | |
|---|---|
| ControlBox | True |
| DrawMode | 13 - Copy Pen |
| DrawStyle | 0 - Solid |
| DrawWidth | 1 |
| Enabled | True |
| FillColor | &H00000000C |
| FillStyle | 1 - Transparent |
| Font | MS Sans Serif |
| FontTransparent | True |
| ForeColor | 000128 |
| HasDC | True |
| Height | 5580 |
| HelpContextID | 0 |
| Icon | (Icono) |
| KeyPreview | False |
| Left | 0 |
| LinkMode | 0 - None |

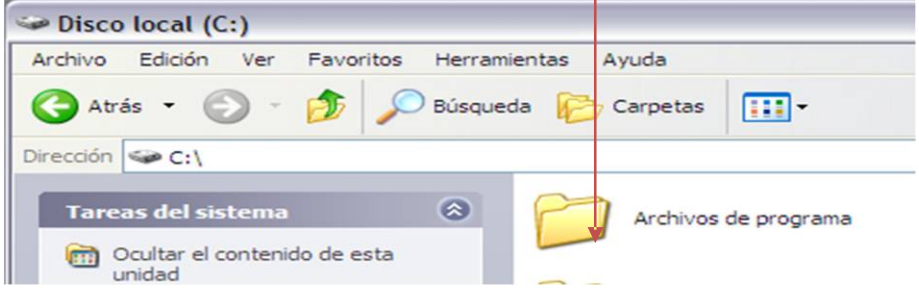
**ForeColor**
Devuelve o establece el color de primer plano usado para mostrar textos y

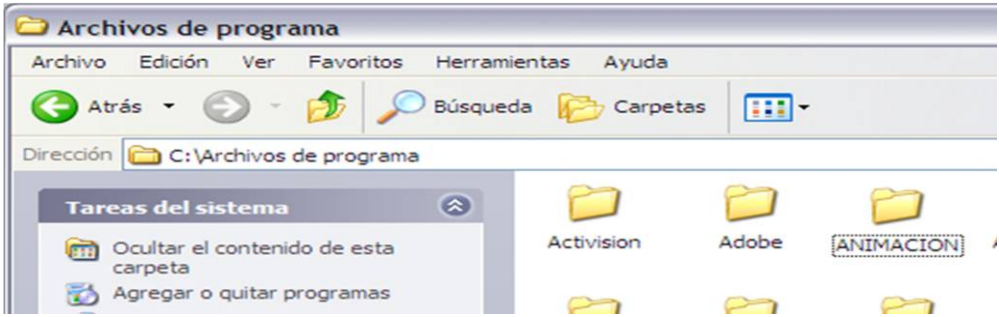
Al final queda así

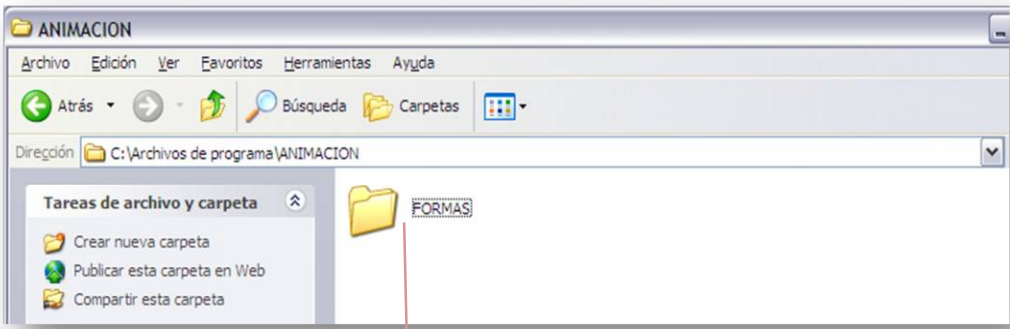



Taller siguiendo las pasos anteriores hacer una mini animación sencilla nota cabe aclarar que para hacer programas en visual se debe crear una carpeta en **mi pc c: archivos de programa con el nombre del software y dentro de ella una que se llame formas hay van las imágenes y sonidos iconos etc. y por fuera los formularios**



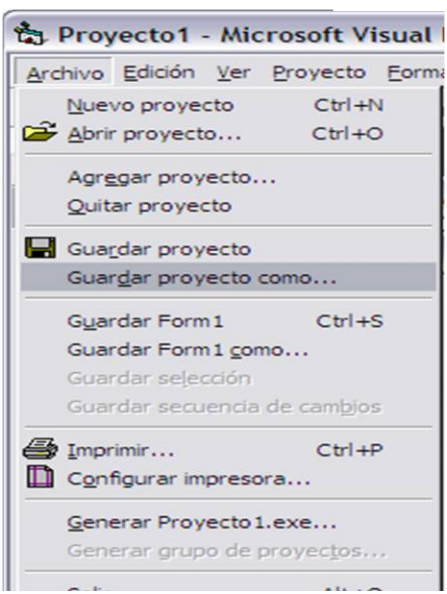



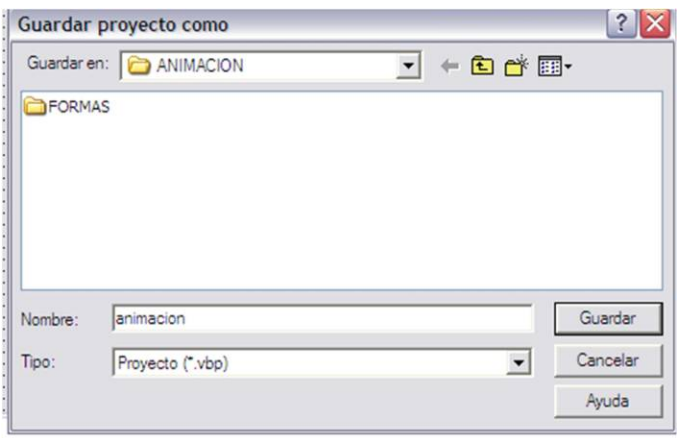


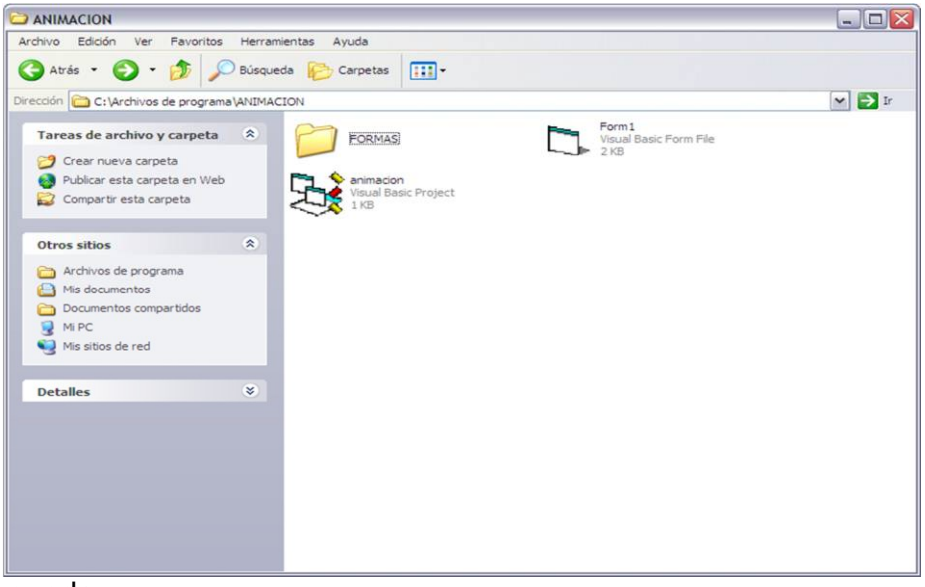




Aquí Iconos Imágenes, Sonidos Etc Que Se Utilizen

## 7. GUARDADO DE PROYECTOS






Así queda

## 8. IMAGE

Propiedades

**Name**

**Caption**

Y para programar visible=true verdadero y false falso, el visible me permite ver o no ver las imágenes en tiempo de ejecución de la aplicación

**Top** - +

**Left**+

-

Para animaciones

# 9. TIMER

Cuando se acaba el tiempo que uno coloco ejecuta las instrucciones

Propiedades

**Interval (tiempo):** el tiempo se da en milisegundos 2000 2 segundos etc.

**Enabled**: habilita y deshabilita el control

## Ejemplo animación con image y timer

Use tres timer y tres image cargue un fondo al formulario con la propiedad picture imágenes todas están en la carpeta principal del programa en archivo de programa dentro de la carpeta formas, imágenes que baje de internet, en la barra de propiedades colocarle al formulario el

caption **animación**

La image uno en las propiedades tienen que estar visible = false

Doble click sobre el timer 1 para programar en el área de programación todo lo que programe se observa pero cada control está separado por una raya al principio y al final

El timer 2 y 3 enabled false en las propiedades y colocarle interval 1000 a los tres timer

Esta área se llama ventana de código fuente, así queda programado luego darle ejecutar

```
Private Sub Timer1_Timer()
Image1.Visible = False
Image2.Visible = True
Timer2.Enabled = True
Timer3.Enabled = False
End Sub

Private Sub Timer2_Timer()
Timer1.Enabled = False
Image2.Visible = False
Image3.Visible = True
Timer3.Enabled = True
End Sub

Private Sub Timer3_Timer()
Image3.Visible = False
Timer1.Enabled = True
Timer3.Enabled = False
Image1.Visible = True
Timer2.Enabled = False
End Sub
```

para iniciar la aplicación

Ejercicio de la guía hacer uno parecido pero con más timer e imágenes

# 10. COMMAND BUTTON

**Propiedades**

**Name**

**Caption**

**Style grafical para boton grafico**

**Picture**

**Tooltiptex:** **mensaje de usuario para botones gráficos**

**Cancel=true funciona con la tecla esc**

**Ejemplo animación con command button timer e image**

Insertar dos imágenes dos botones de comando y cambiarles el caption de acuerdo a la imagen para sacar la famosa tecla nemometríca que se activa con alt y la letra subrayada se coloca en el caption del botón un & arpensen antes de la letra deseada

Los dos timer enabled false

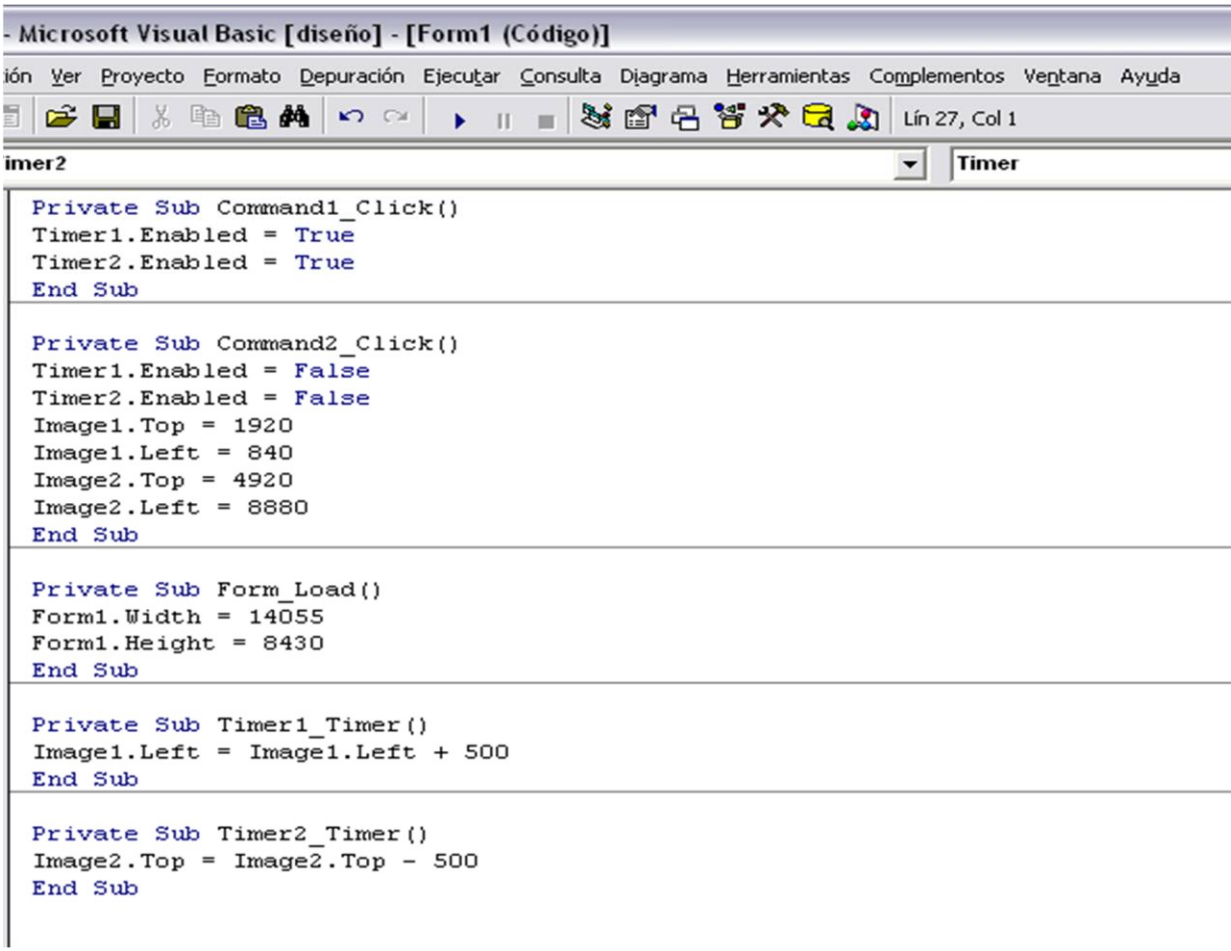
Interval 1000 en los dos timer, doble click en los dos timer para programar y en los command button, al reducir el interval del el recorrido de las imágenes van más rápidamente

Image1.top = 1920

Estos valores son de las images en la barra de propiedades top y left significa en donde y que posición se encuentran las imágenes, esto se hace para que al darle click en el botón detener la imagen quede en el mismo lugar donde estaban antes de ejecutar la animación

El width y height es para que cuando se modifique el tamaño del formulario en programación no se modifique en ejecución esto se programaría en el load del formulario

**Queda así**



Ejercicio hacer uno parecido al de la guía pero con más imágenes

## 11. TIPOS DE VARIABLE 2

**Private solo para un módulo o formulario**

**Public para todo los módulos**

**Dim en un solo formulario**

**Static procedimiento dentro de un formulario**

## 12. TIPO DE VARIABLES 3

**Integer de tipo números enteros**

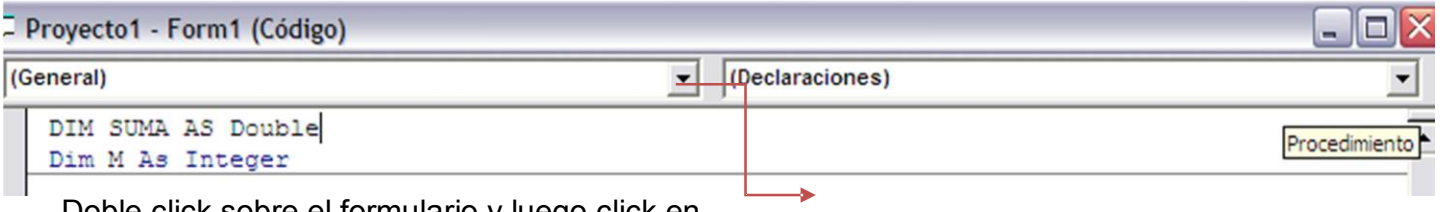
**String cadena de caracteres “”**

**Long entero largo**

**Double para números grandes**

**Variant para cualquier valor**

**Las variables se definen en general declaraciones**




Doble click sobre el formulario y luego click en

## 13. TEXTBOX CAJAS DE TEXTO

**Propiedades**

**Enabled=impide que el usuario entre a la caja y cambie algún valor**

**Text esta es la propiedad de colocar texto**

**Name**

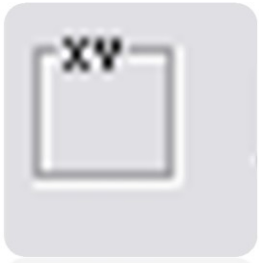
**Setfocus dar foco con enter esto se programa en el evento keypress**

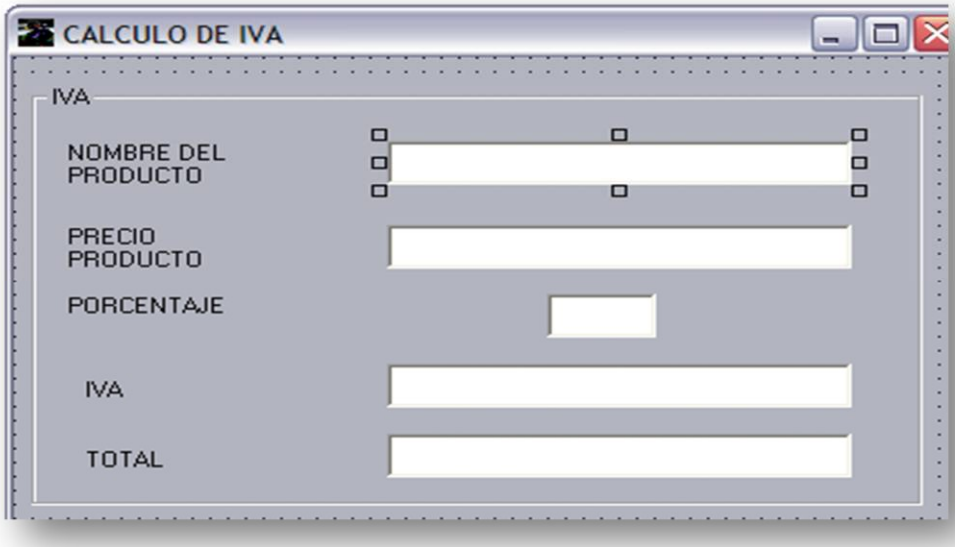
**Multilinea para poder dar enter y seguir escribiendo**

**Scrollbars, barras de desplazamiento en las cajas**

**Ejemplo cálculo del iva**

**Frame control que contiene en un formulario otros controles**



**La programación queda así**

- Microsoft Visual Basic [diseño] - [Form21 (Código)]

ición Ver Proyecto Formato Depuración Ejecutar Consulta Diagrama Herramientas Complementos Ventana Ayuda

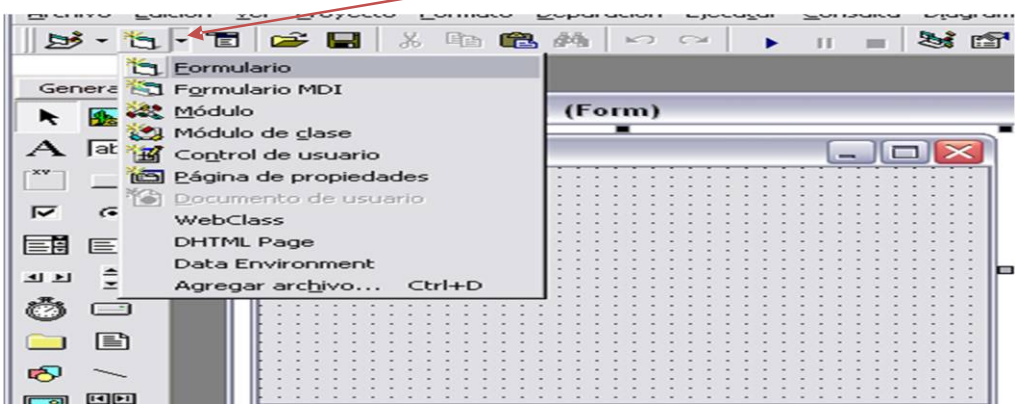
Lín 3, Col 1

Form | Activate

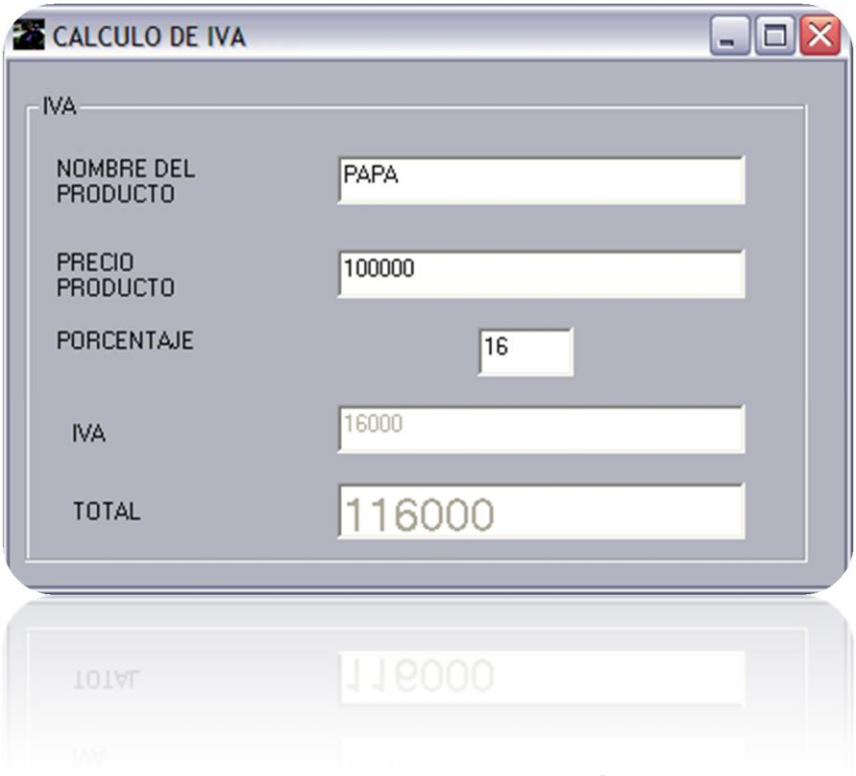
```
Dim PRECIO, PORCENTAJE, iva, total As Double
Private Sub Form_Activate()
Txnom.SetFocus
End Sub

Private Sub Txnom_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Txprecio.SetFocus
End If

End Sub

Private Sub Txporcent_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
PORCENTAJE = Val(Txporcent.Text)
iva = PRECIO * PORCENTAJE / 100
Txiva = iva
total = PRECIO + iva
Txtotal = total
End If
End Sub

Private Sub Txprecio_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
PRECIO = Val(Txprecio.Text)
Txporcent.SetFocus
End If
End Sub
```

SIGNIFICA SALTA A OTRA CAJA AL DAR ENTER

**If** significa si **else** significa no **val** significa convertir el valor por defecto de una caja de texto a valor numérico, para cargar un formulario al mismo proyecto sin necesidad de abrir uno nuevo se hace de la siguiente manera dentro del proyecto se da click aquí y luego, **keyascci=** 13 significa el valor de la tecla enter en la tabla de ascci



En tiempo de ejecución el mini programa queda así



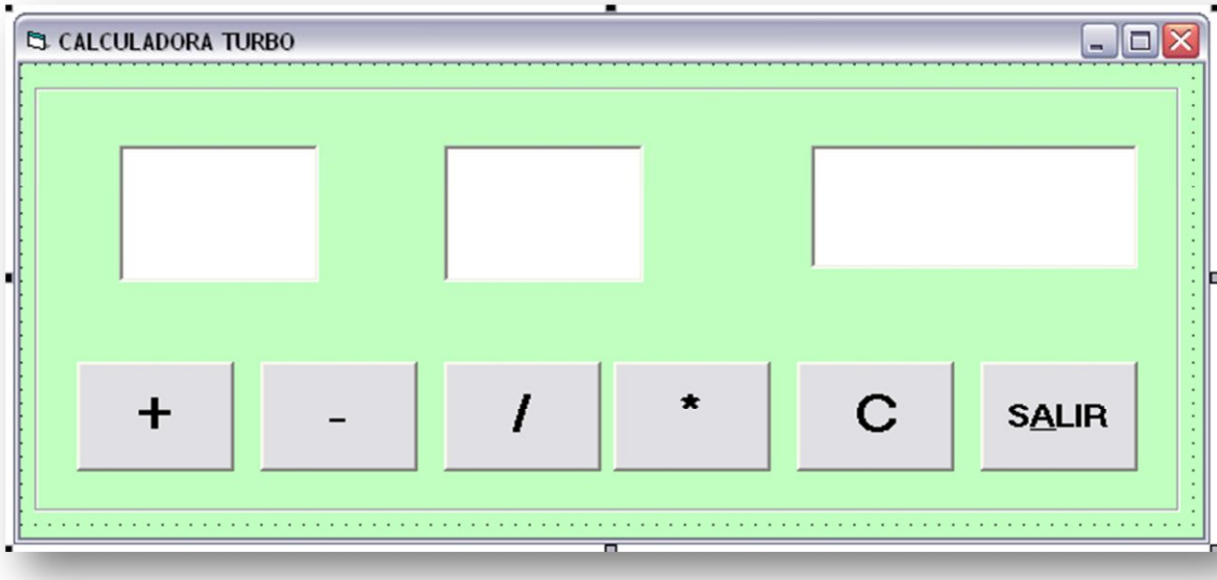

El ejercicio es hacer uno parecido y también utilizando command button pero de comisiones siguiendo el ejemplo anterior

## 14. TIPOS DE INTERFAZ

Las aplicaciones tipo Windows no tiene las mismas interfaces con este tipo de documento.

1. **Sdi Interfaz De Documento Único(Simple Document Interface):** En Este Tipo De Interfaz Solo Se Abre Un Documento Y Cuando Abre Una Ventana Debe Cerrar La Anterior, Ejemplo ,El Calendario, La Calculadora ,Bloc De Notas Y Se Utiliza Cuando Se Hace Una Sola Aplicación A La Vez

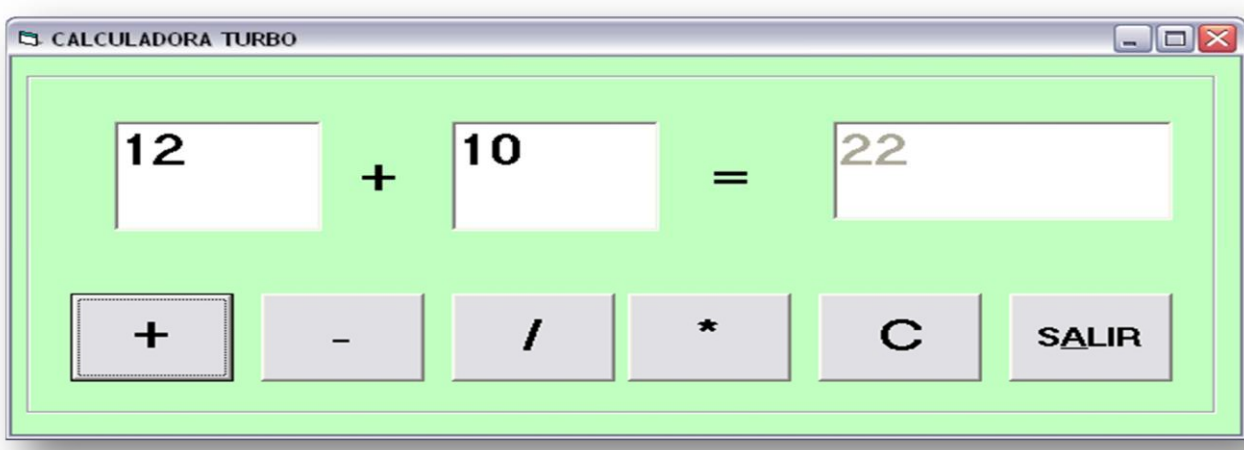
Ejemplo para este ejercicio se utilizó cajas de texto labels command button frame, los labels



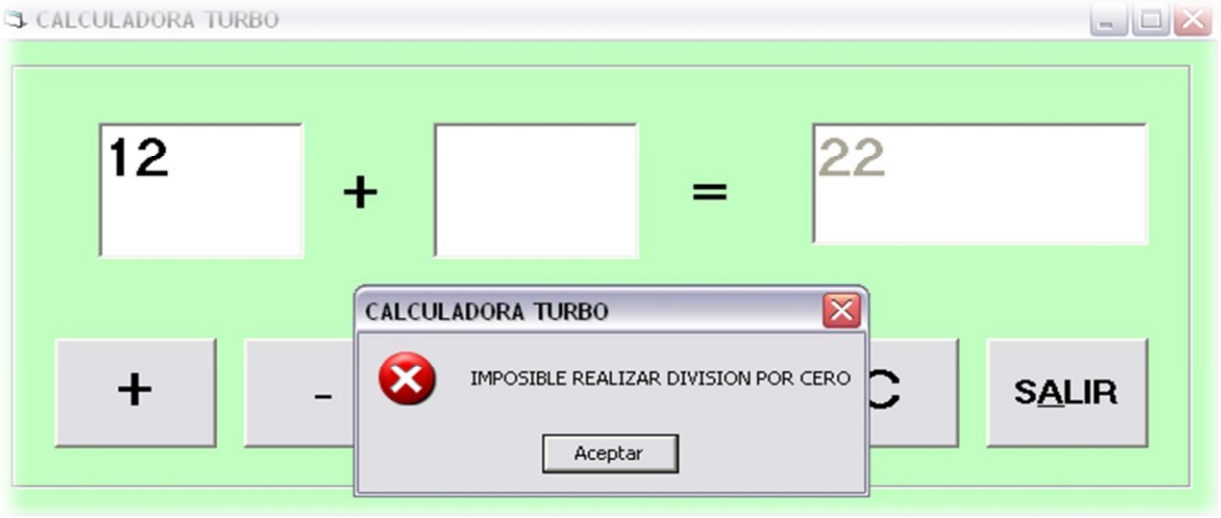

deben estar de tipo transparente en **Backstyle** agrandar la fuente de las cajas en font y colocarle color.

En este ejercicio se utilizan los operadores booleanos que recordemos cuales son el **or** significa o y el **and** significa y recordemos que el **if** significa si **else** significa no **end if** significa fin del sí y la propiedad **val** de las cajas de texto es para convertir el valor por defecto de texto a valor numérico el **then** significa entonces y **las comillas dobles pegadas** significan vacío **msgbox** significa mensaje de usuario el cual trae 3 opciones **vbcritical** que significa error **vbinformasion** significa información y **vbexclamacion**, recuerden que la caja de resultados va **enabled** false para evitar que el usuario cambie un resultado

A continuación mostrare detalladamente el código fuente de esta aplicación para que les



sirva como modelo para hacer otras aplicaciones parecidas en tiempo de ejecución se vería así




El código fuente de la siguiente aplicación es:

Command4 | Click

```
Dim MUL As Double
Dim DIV As Double
Dim RESTA As Double
Dim SUMA As Double
Private Sub Command1_Click()
Label1.Caption = "+"
Label2.Caption = "="
SUMA = Val(Text1.Text) + Val(Text2.Text)
Text3.Text = SUMA
End Sub
Private Sub Command2_Click()
Label1.Caption = "-"
Label2.Caption = "="
RESTA = Val(Text1.Text) - Val(Text2.Text)
Text3.Text = RESTA
End Sub

Private Sub Command3_Click()
On Error Resume Next
If Err.Number = 13 Then
End If
If Text1.Text = "" Or Text2.Text = "" Or Text1.Text = 0 Or Text2.Text = 0 Then
MsgBox "IMPOSIBLE REALIZAR DIVISION POR CERO", vbCritical, "CALCULADORA TURBO"
Text1.Text = ""
Text2.Text = ""
Text1.SetFocus
Else
Label1.Caption = "/"
Label2.Caption = "="
DIV = Val(Text1.Text) / Val(Text2.Text)
Text3.Text = DIV
End If
End Sub
```

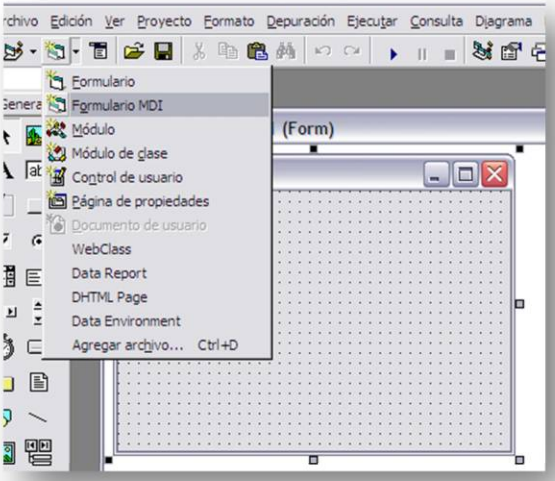
```
Private Sub Command4_Click()
Label1.Caption = "X"
Label2.Caption = "="
MUL = Val(Text1.Text) * Val(Text2.Text)
Text3.Text = MUL
End Sub

Private Sub Command5_Click()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text1.SetFocus
End Sub

Private Sub Command6_Click()
End
End Sub

Private Sub Form_Activate()
Text1.SetFocus
End Sub

Private Sub Form_Load()
Form1.Height = 4710
Form1.Width = 10170
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text2.SetFocus
End If
End Sub
```

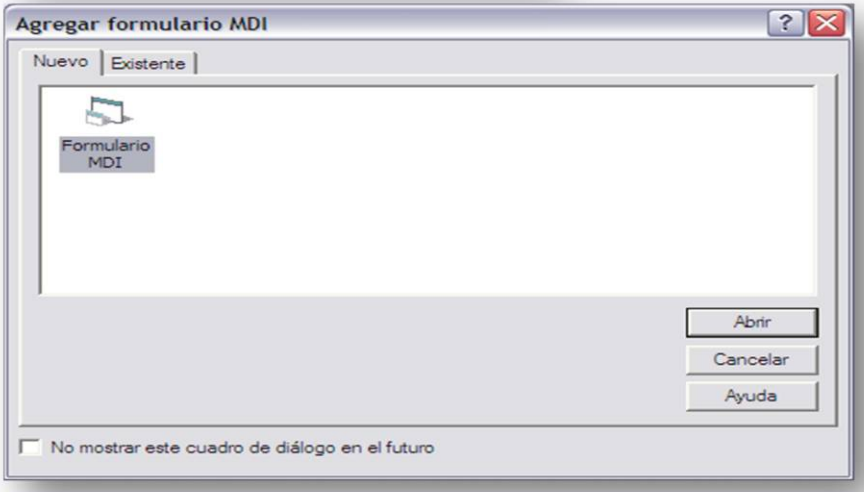
Ejercicio de la guía es hacer una calculadora parecida con otros fondos y entorno grafico

Mdi(multiple documento interface) interfaz de múltiple documento): permite crear una aplicación que contenga varias ventanas o formularios dentro de otro formulario denominado formulario contenedor ejemplo Word Excel, power point, etc.

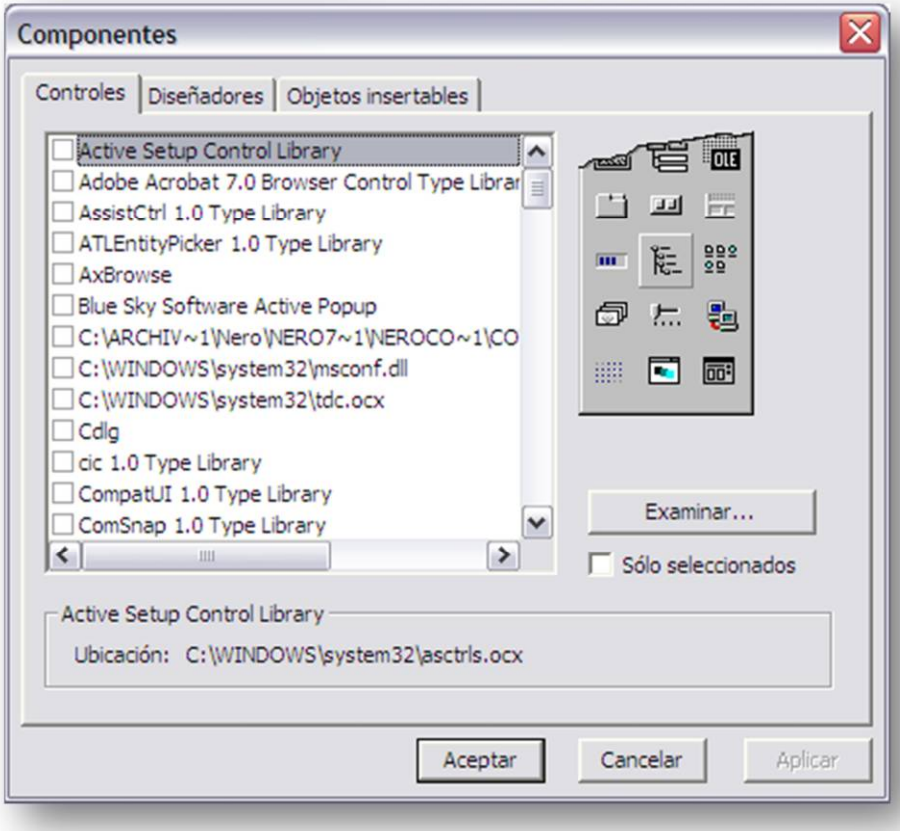
Ejemplo de este es una aplicación demo de notas la carpeta se encuentra elaborada en c archivos de programa

Para sacar el mdi de un proyecto es así



Para elaborar la **toolbar** es decir la barra de herramientas la **statusbar** es decir la barra de estado, los dbcombo el data grid los adocd que son conectores a la base de datos junto a los data, hay que sacar algunos componentes de visual Basic 6.0 así:



MICROSOFT ADO DATA CONTROL 6.0 (OLEDB)

MICROSOFT  DATA BOUND LIST CONTROLS 6.0

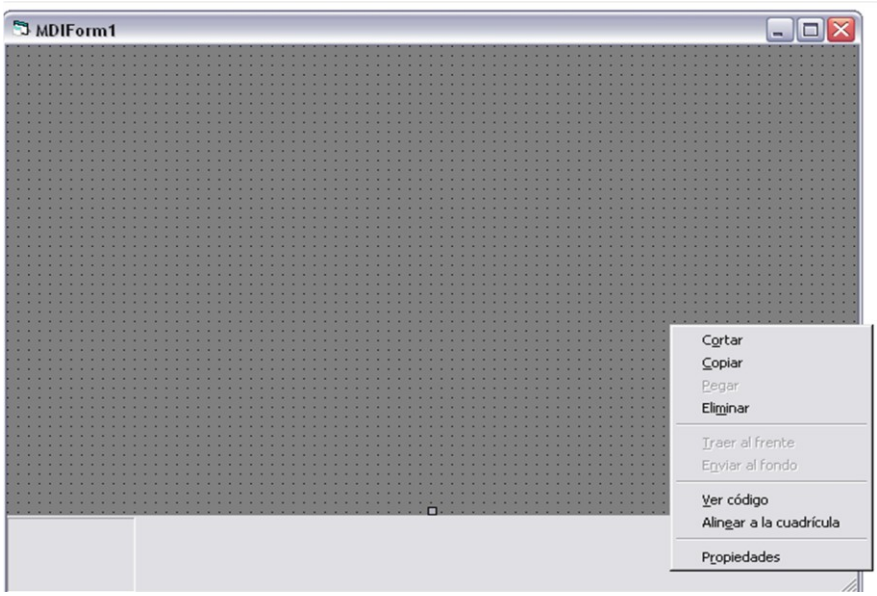
MICROSOFT DATA GRID CONTROL 6.0 (OLEDB)

MICROSOFT WINDOWSCOMMON CONTROLS 6.0 (SP6)

Automáticamente la barra de controles se actualizara así

AQUÍ PARA CREAR LA BARRA DE ESTADO

Se dibuja en el mdi  y después aparecerá esto click derecho sobre la statusbar y propiedades

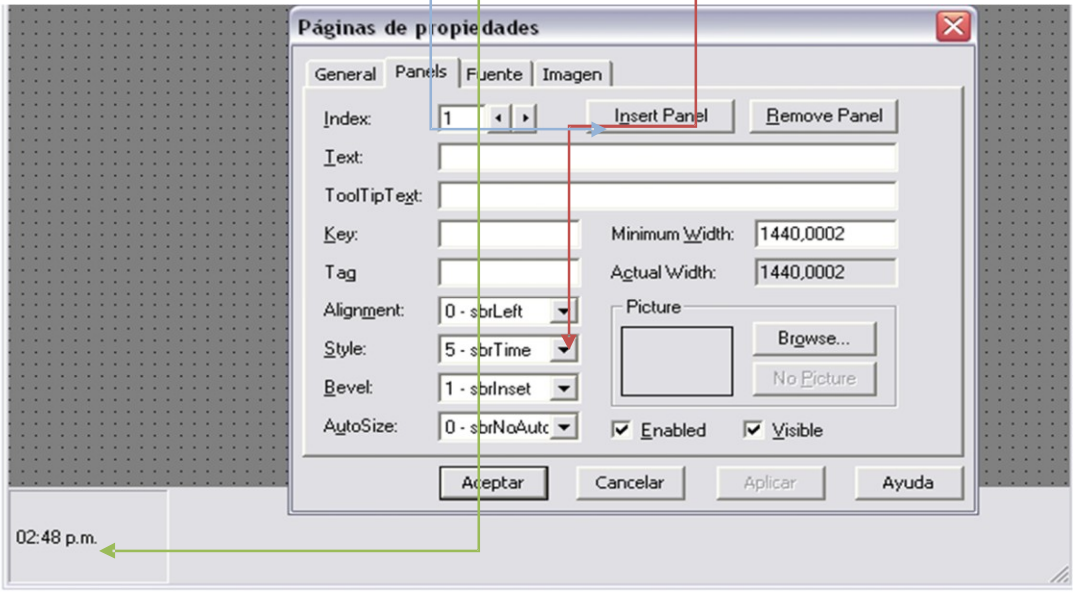



Después de darle propiedades sale esto

Para que aparezca el estilo

Inserta un panel

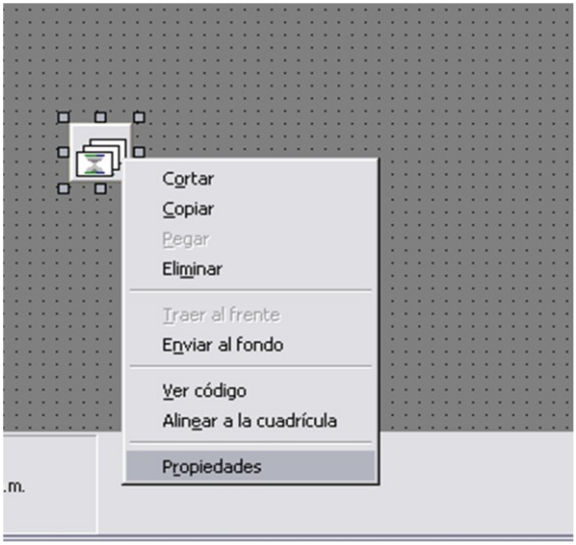
se configura el estilo



Para crear la toolbar es decir la barra de herramientas es así

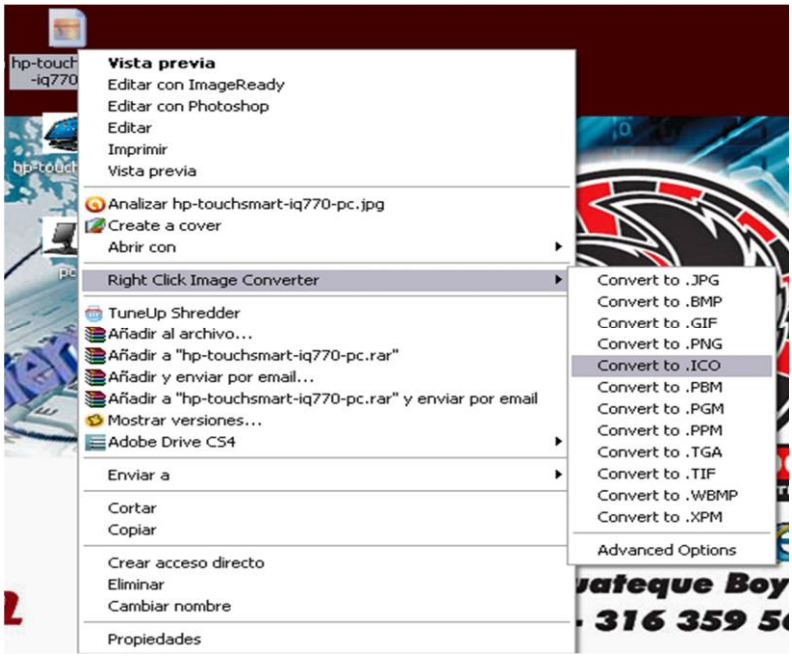
SE DIBUJA EL IMAGELIST PARA AQUÍ CARGAR LOS ICONOS QUE VOY A UTILIZAR EN LOS BOTONES

Click derecho sobre el imagelist y sale esto, para hacer esto los iconos a utilizar ya deben estar en la carpeta formas dentro de la carpeta principal del software

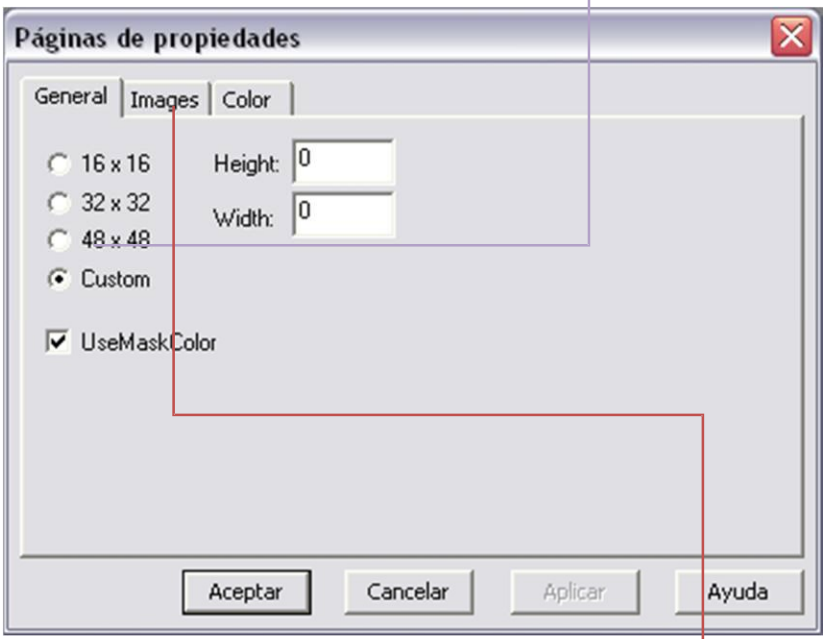



## 15. CON EL RIGHT CLICK IMAGE CONVERTER

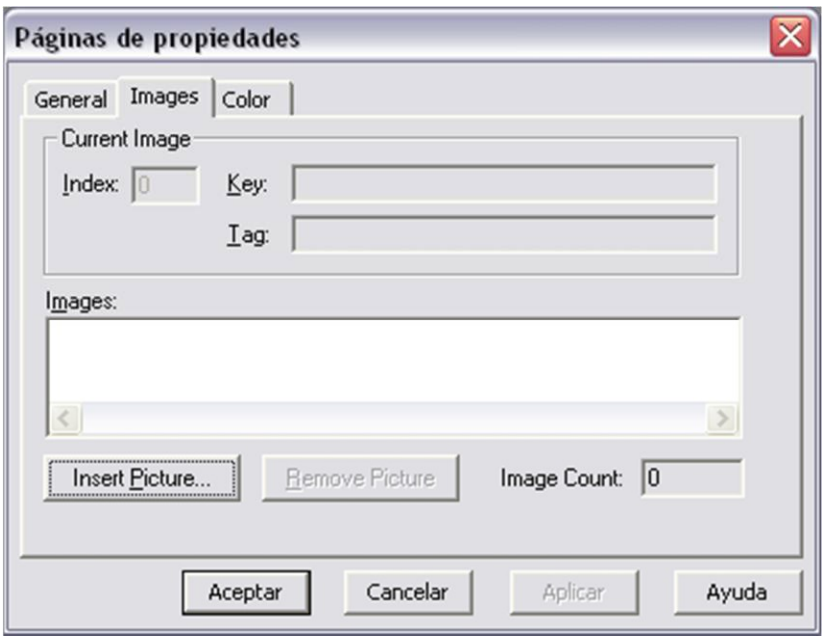
Teniendo este programa instalado en la pc se le cambia el formato a las imágenes convirtiéndolas a iconos tan solo con darle click derecho tal como muestro a continuación

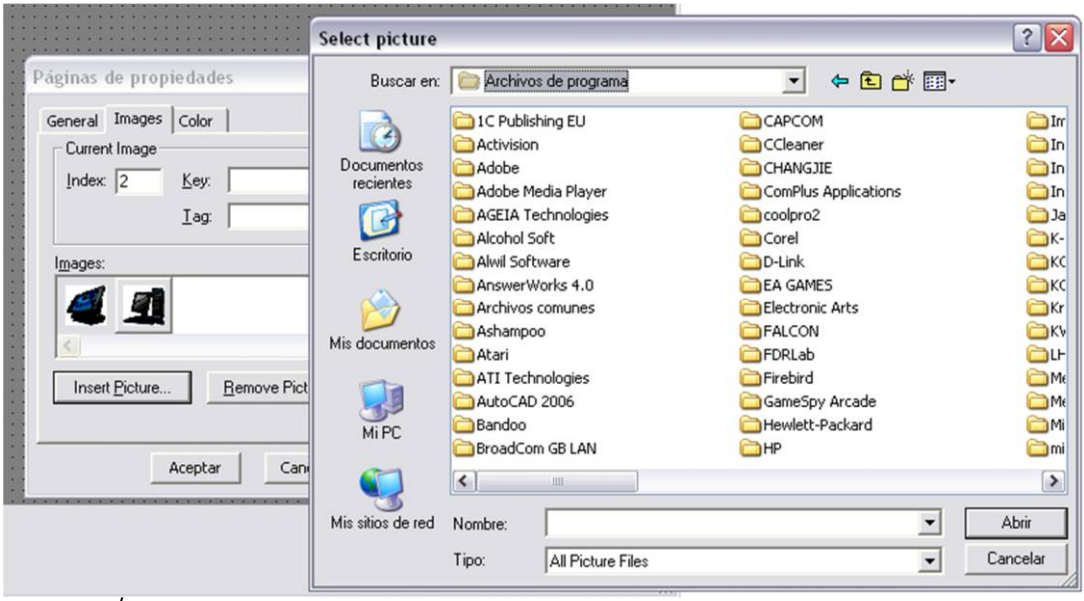


Ya teniendo los iconos guardarlos en la carpeta anteriormente mencionada después se la da click derecho al imagelist en visual y sale este cuadro, elegir iconos de 32 x 32 o 48 x 48




Luego click en esta pestaña

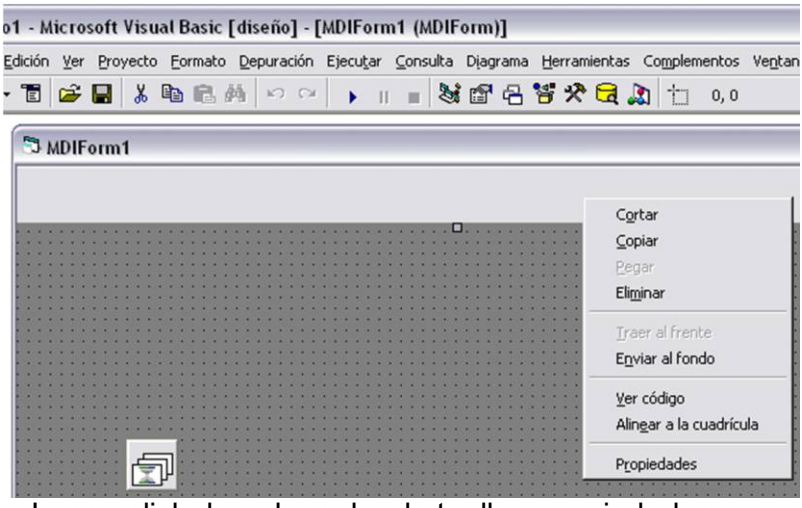




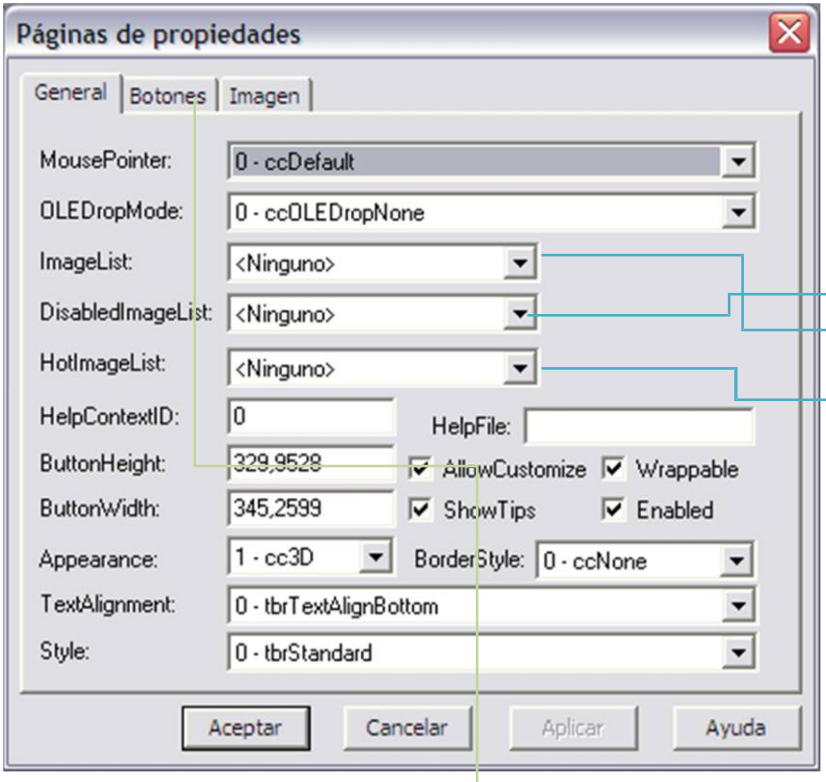

Luego click aquí para adherir los iconos

Queda así y luego aceptar luego proceder a crear la toolbar

TOOLBAR

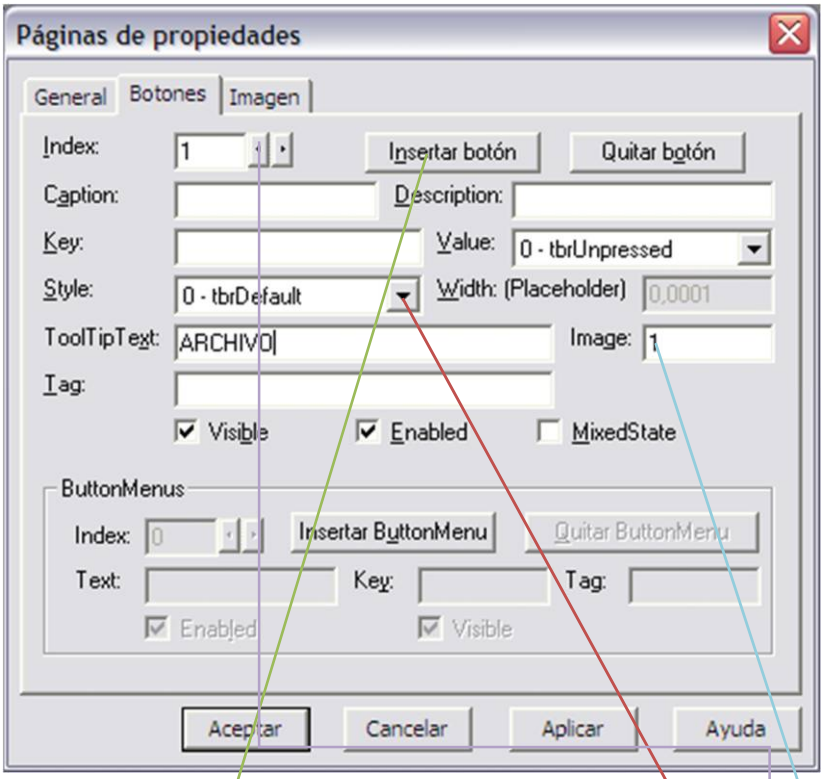



Luego click derecho sobre la toolbar propiedades




CARGAR AQUÍ LA IMAGELIST

CLICK ACA PARA CARGAR LOS BOTONES



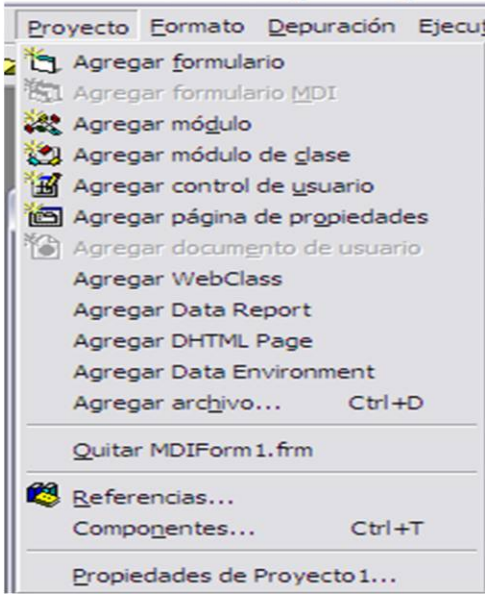

AQUÍ PARA INSERTAR BOTON

AQUÍ PARA COLOCAR EL NÚMERO DE LA IMAGEN

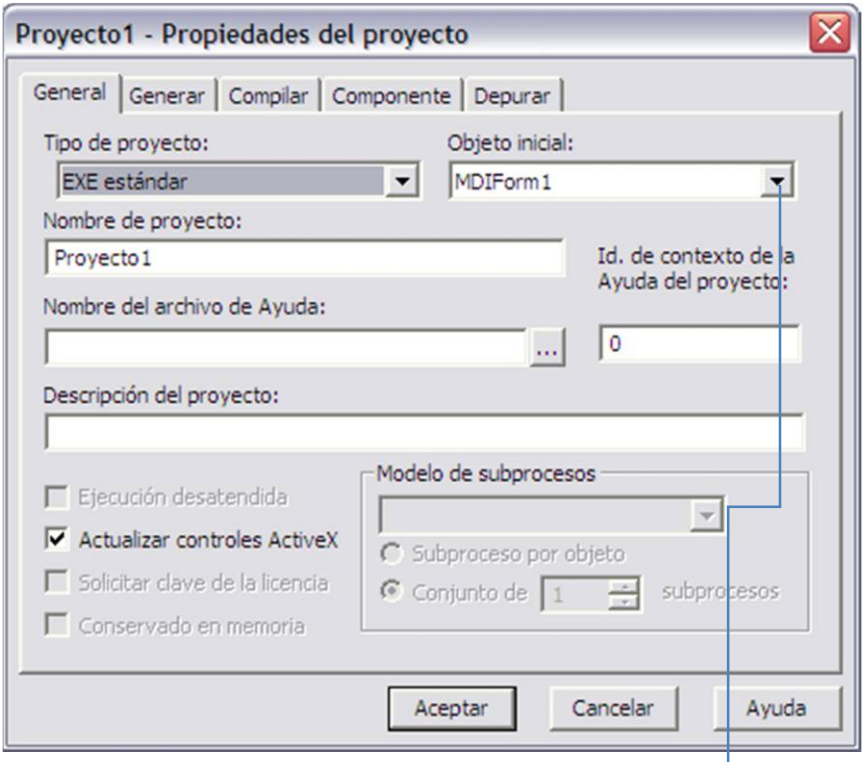
SI SE QUIERE HACER UN BOTON SEPARADOR CAMBIAR EL STYLE AQUÍ COMO TIPO TERSEPARATOR CABE ACLARAR QUE POR CADA BOTON HAY QUE CAMBIAR EL INDEX

Al Finalizar Darle Aplicar Y Aceptar Y Quedara Asi

Cuando hay varios formularios en el proyecto para probarlo se entra aquí para cambiar la ejecución del formulario dado



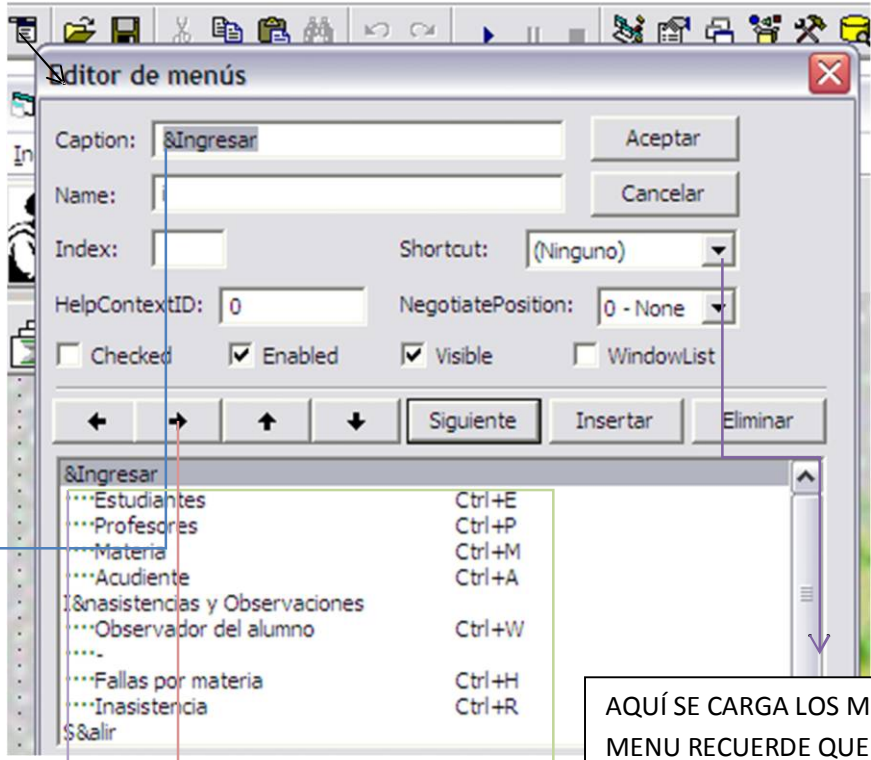
Click en el menú proyecto, luego propiedades de proyecto y sale este otro cuadro




SELECCIONAR EL FORMULARIO QUE DESEA CARGAR DEL PROYECTO PARA EJECUCION

## *16. CREACION DE MENUS*

Para La Creacion *De Los Menus Entramos A la Siguiente Opción*

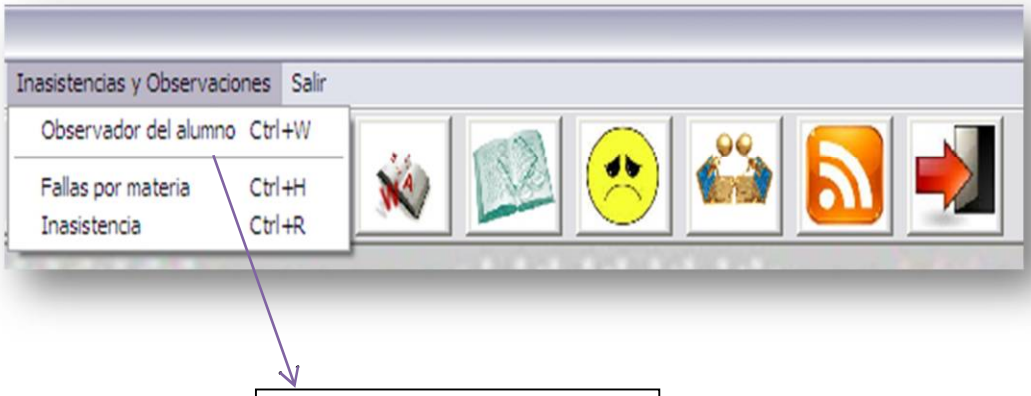



AQUÍ SE CARGA LOS METODOS ABRVIADOS DEL MENU RECUERDE QUE UN & ARPENSEN ANTES DEL NOMBRE CREA LA FAMOSA TECLA NEMOMETRICA ES DESIR LA LETRA SUBRAYADA DE CADA MENU QUE FUNSIONA CON ALT Y LA LETRA SUBRAYADA

NOMBRE DEL MENU

MENU RPRINCIPAL

Cuatro puntos significa hijo del menú principal y ocho puntos significa hijo del hijo del menú principal es decir crea sub menús una raya al piso crea un separador entre cada menú para sacar los hijos utilizamos estas flechas estando listo queda así




SEPARADOR

Carga de formularios por medio de los menús

Se da doble click al nombre del menú y se programa esto de acuerdo al formulario que se vaya a cargar se coloca el name del formulario conél.**show** que significa cargar formulario

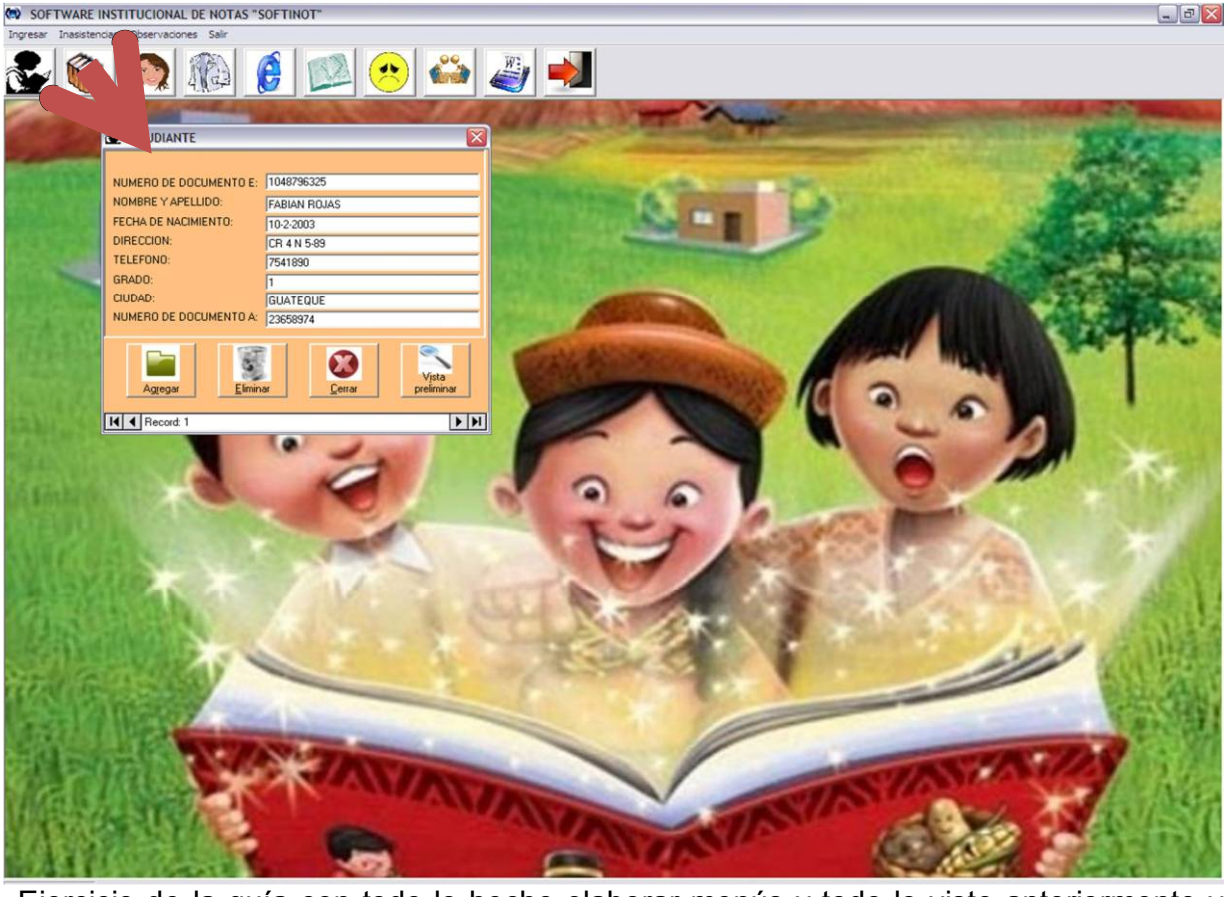
```
Click

Private Sub e_Click()
frmESTUDIANTE.Show
End Sub

Private Sub l_Click()
Form3.Show
End Sub
```

Para Programar La Toolbar Ósea La Barra De Herramientas Los Nombre Que A Continuación Muestro Deben Ser El Nombre Que Se Le Dio Al Menú Donde Se Carga El Formulario

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1: Call e_Click
Case 3: Call n_Click
Case 5: Call a_Click
Case 7: Call p_Click
Case 9: Call t_Click
Case 11: Call l_Click
Case 13: Call u_Click
Case 15: Call o_Click
Case 17: Call m_Click
Case 19: Call r_Click
End Select
End Sub
```

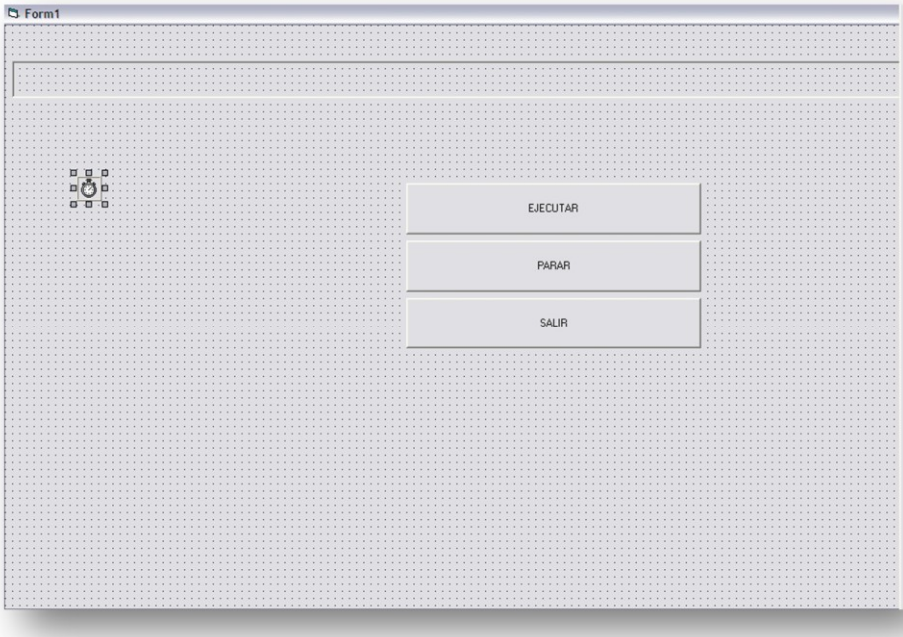
**El select** es según el **index** que se creó al crear los botones. **Call** significa llamar formulario con click, **case** significa número del menú y formulario que voy a cargar de acuerdo a la barra de herramientas anteriormente diseñada, el fondo del mdi se carga en la propiedad **picture**

Al final el mdi terminado quedara así

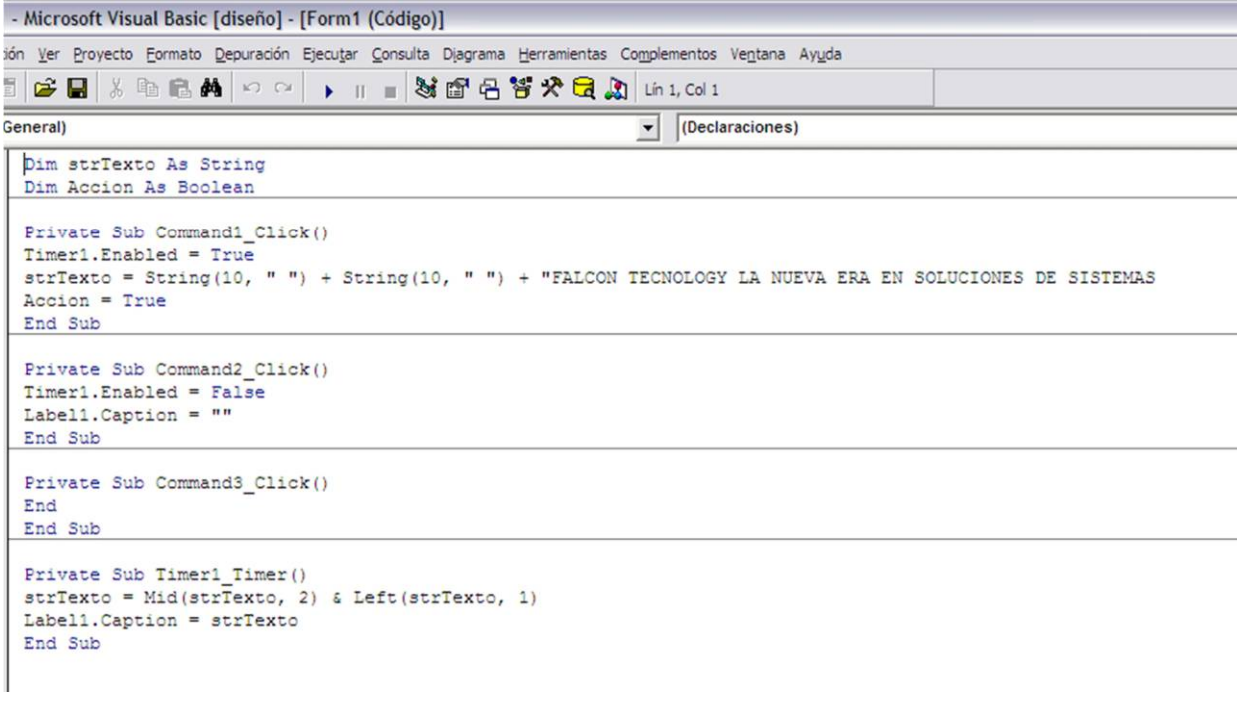



Ejercicio de la guía con todo lo hecho elaborar menús y todo lo visto anteriormente y hacer que funcionen los botones y los menús cargando varios formularios

# 17. TEXTO CON MARQUESINA

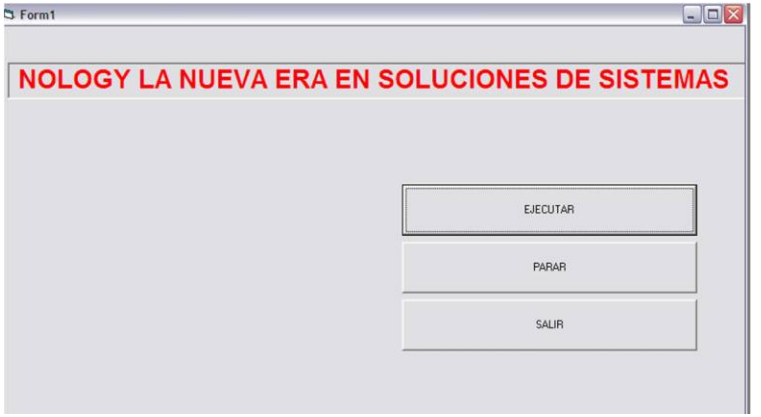



En este ejemplo utilizamos cajas de texto y command button y timer, a continuación muestro el ejemplo de programación, el interval del timer fue 100

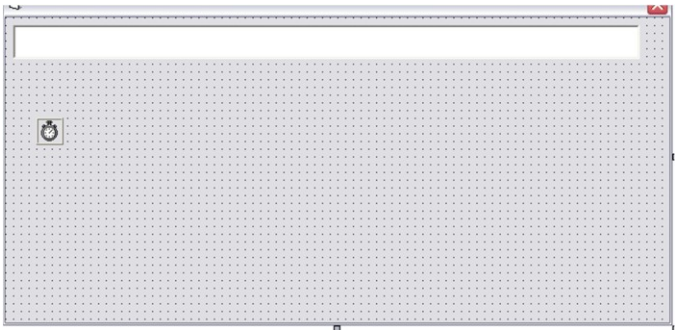


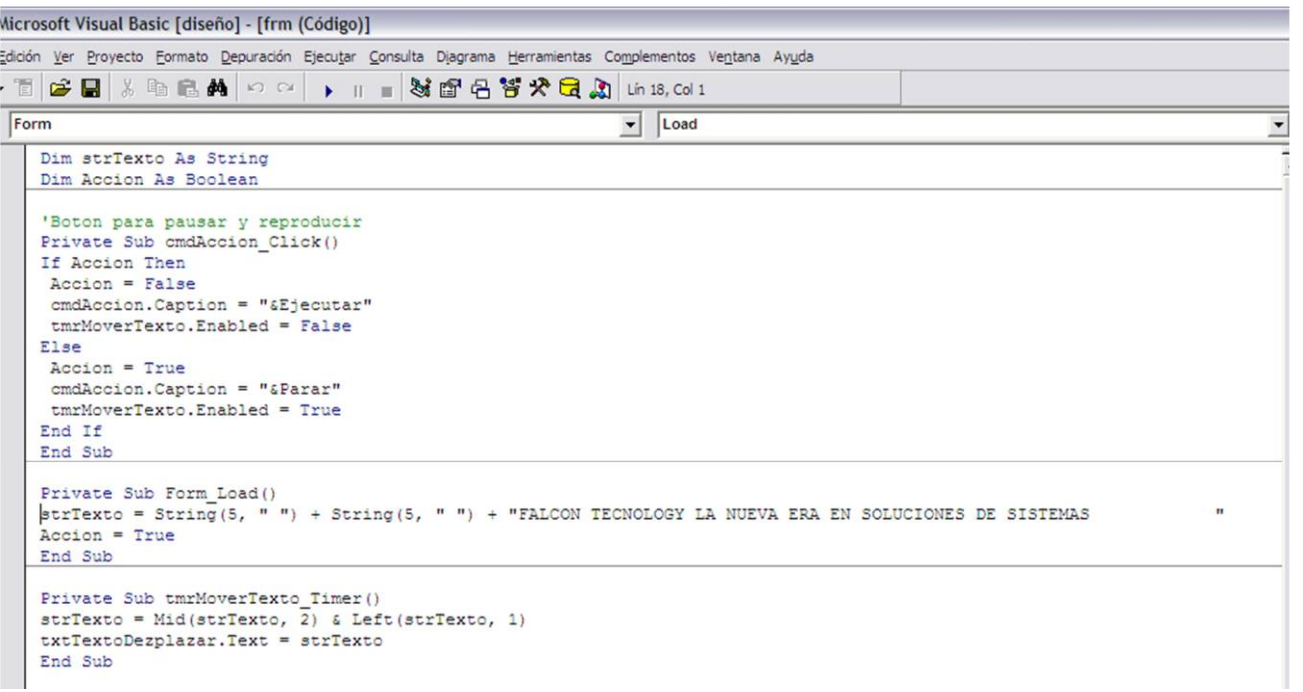
**End** significa fin salir, la función **mid & left** es la velocidad del texto

Así queda en ejecución



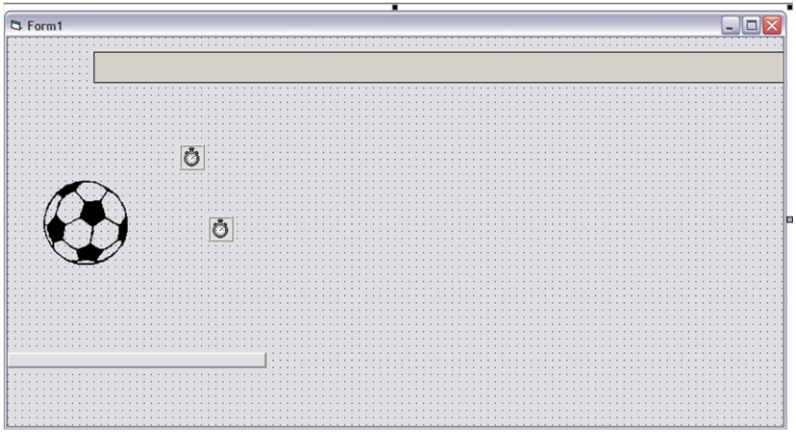

Para hacerlo de manera automática apenas empiece la ejecución quedaría así






```
Dim strTexto As String
Dim Accion As Boolean

'Boton para pausar y reproducir
Private Sub cmdAccion_Click()
If Accion Then
 Accion = False
 cmdAccion.Caption = "&Ejecutar"
 tmrMoverTexto.Enabled = False
Else
 Accion = True
 cmdAccion.Caption = "&Parar"
 tmrMoverTexto.Enabled = True
End If
End Sub

Private Sub Form_Load()
strTexto = String(5, " ") + String(5, " ") + "FALCON TECNOLOGY LA NUEVA ERA EN SOLUCIONES DE SISTEMAS           "
Accion = True
End Sub

Private Sub tmrMoverTexto_Timer()
strTexto = Mid(strTexto, 2) & Left(strTexto, 1)
txtTextoDezplazar.Text = strTexto
End Sub
```

**Load** significa en tiempo de carga y **actívate** significa en tiempo de activación estos son los famosos **metodo** ya sea al hacer click con el mouse presionar una tecla etc.




## 18. ANIMACION CON REBOTE DE BALON

La programación quedara así

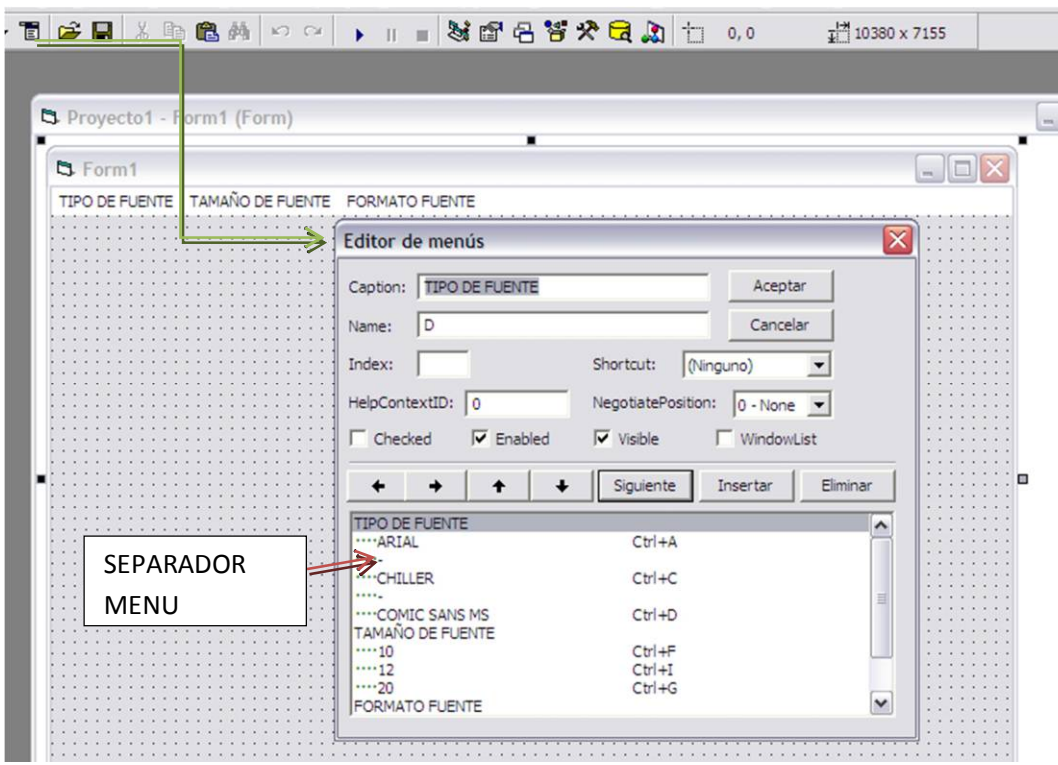
(General) (Declaraciones)

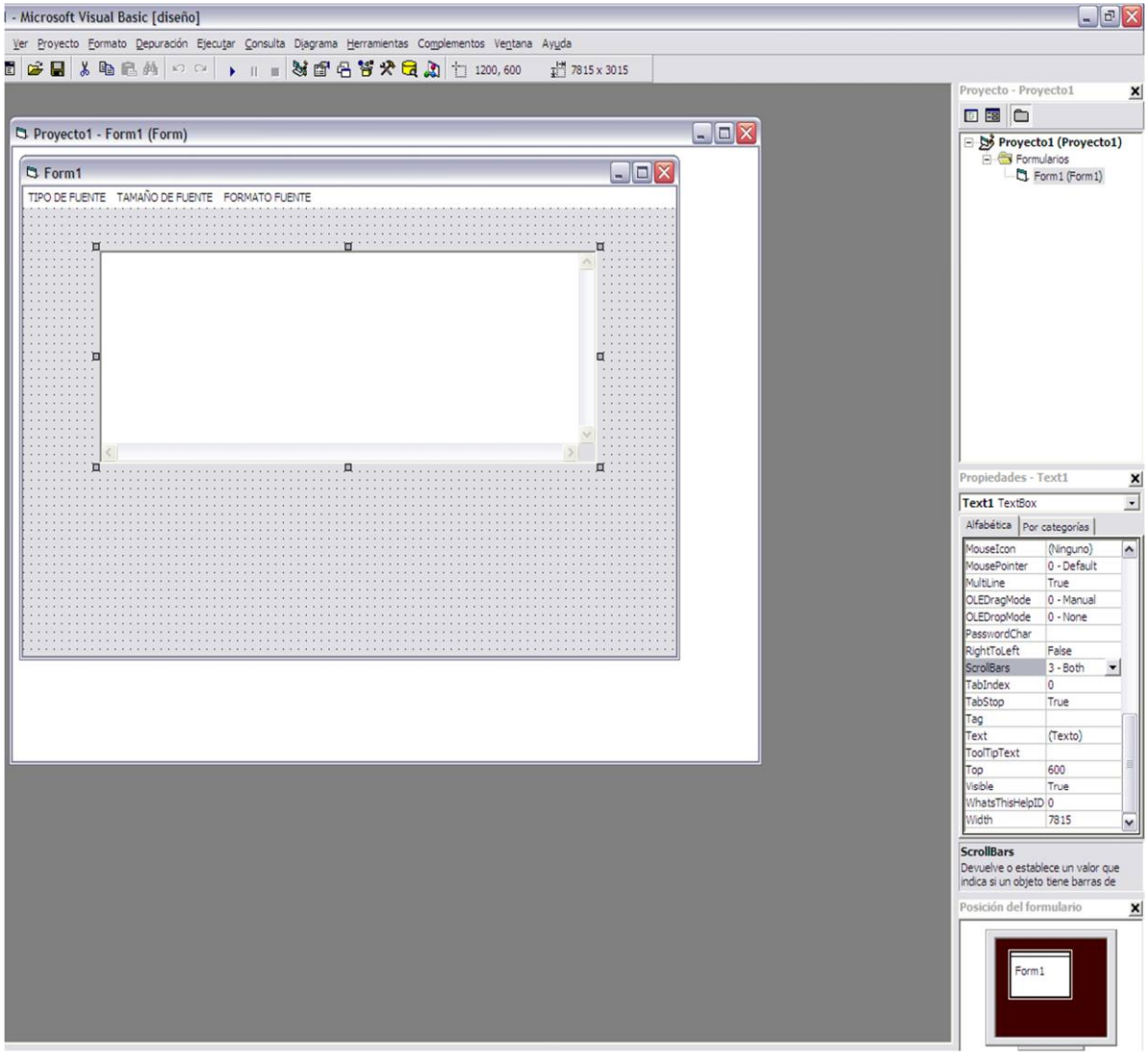
```
Dim strTexto As String
Dim Accion As Boolean
Dim Fuerza As Double
Dim Velocidad As Double
Private Sub Form_Load()
strTexto = String(5, " ") + String(5, " ") + "FALCON TECNOLOGY LA NUEVA ERA EN SOLUCIONES DE SISTEMAS            "
Accion = True
'Posicionamos el picture o el control image en el form
    Image1.Top = 10
    Image1.Left = 10

    ' maximiza el formulario
    Me.WindowState = 2

    'Establecemos un valor al Timer y lo activamos
    Timer1.Enabled = True
    Timer1.Interval = 10

    'Establecemos valores a las variables _
     para el efecto de rebote
    Fuerza = 1.9
    Velocidad = 0
End Sub
Private Sub Timer1_Timer()
 If Image1.Top > 4000 Then
        ' Esta línea se ejecuta cuando llega al limite
        Image1.Top = 4000
        Velocidad = Velocidad * -0.9
    End If

    Velocidad = Velocidad + Fuerza

    ' mueve EL top en el control image
    Image1.Top = Image1.Top + Velocidad
End Sub
Private Sub Timer2_Timer()
strTexto = Mid(strTexto, 2) & Left(strTexto, 1)
txtTextoDezplazar.Text = strTexto
End Sub
```

Ejercicio de la guía elaborar un efecto marquesina de una forma parecida a las anteriores

## 19. EJEMPLO DE EJERCICIO DE UN MINI PROCESADOR DE TEXTO

La elaboración de los menús quedaría así

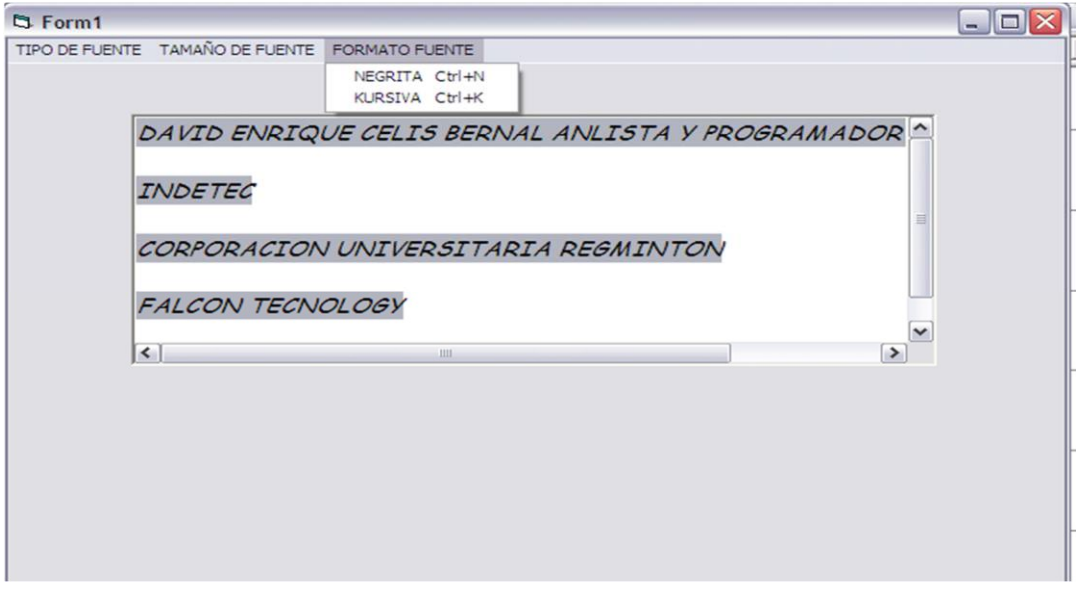





En las propiedades de la caja de texto en la opción multilinea que quede true y activar la propiedad **scrollbars – both** la programación de este ejercicio es la siguiente

```
Private Sub A_Click()
Text1.Font.Size = 20
End Sub

Private Sub EE_Click()
Text1.Font.Bold = True
End Sub

Private Sub H_Click()
Text1.FontName = "Chiller"
End Sub

Private Sub I_Click()
Text1.Font.Size = 10
End Sub

Private Sub N_Click()
Text1.FontName = "Comic Sans MS"
End Sub

Private Sub P_Click()
Text1.Font.Size = 12
End Sub

Private Sub R_Click()
Text1.FontName = "Arial"
End Sub

Private Sub TT_Click()
Text1.Font.Italic = True
End Sub
```

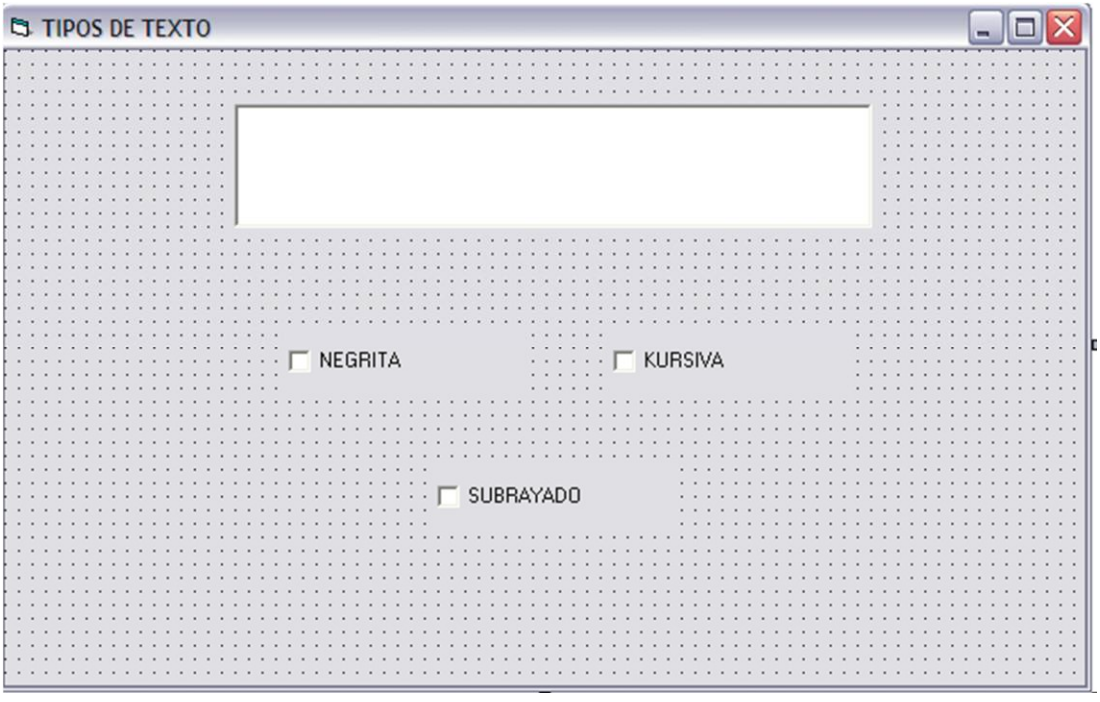
La propiedad **font.size** significa tamaño de fuente la propiedad **fontname** significa tipo de fuente o nombre de la letra estos nombres se sacan del procesador de **texto Microsoft Word** y se copia y se pegan tal cual como se llaman la propiedad **font.bold** significa negrita y la propiedad **font.italic** significa cursiva en tiempo de ejecución esta aplicación queda así



El ejercicio de la guía es hacer otro parecido utilizando más tipos de fuentes y tamaños de fuente

## 20. CHECKBOX Y OPTION BUTTON

Para estos ejemplos voy a hacer una aplicación de negrita cursiva y subrayado y un ejemplo de aplicar colores

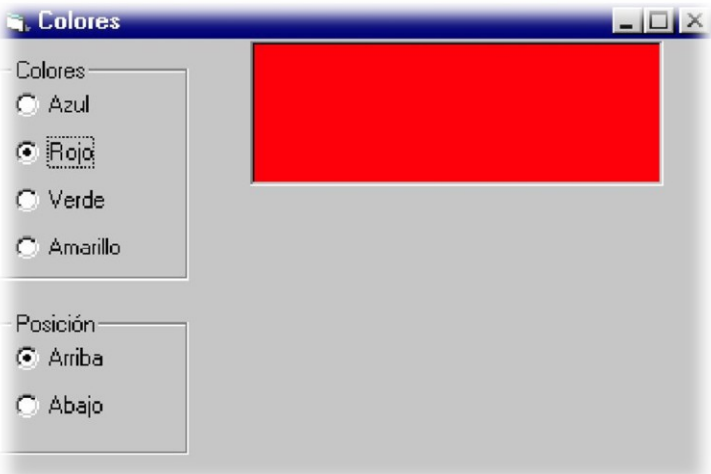



El código fuente de esta aplicación es el siguiente

```
Private Sub Check1_Click()
If Check1.Value = vbChecked Then
Text1.Font.Bold = True
Else
Text1.Font.Bold = False
End If
End Sub

Private Sub Check2_Click()
If Check2.Value = vbChecked Then
Text1.Font.Italic = True
Else
Text1.Font.Italic = False
End If
End Sub

Private Sub Check3_Click()
If Check3.Value = vbChecked Then
Text1.Font.Underline = True
Else
Text1.Font.Underline = False
End If
End Sub
```

La opción **font.underline** significa subrayado la opción **vbchecked** significa que la casilla de verificación o **checkbox** está activado
El ejercicio de la guía hacer uno parecido para aplicar lo visto en el ejemplo

## 21. SENCILLO PROGRAMA DE COLORES Y POSICIONES




En la figura se muestra el formulario y los controles de un ejemplo muy sencillo que permite mover una caja de texto por la pantalla, permitiendo a su vez representarla con cuatro colores diferentes.

En la tabla se describen los controles utilizados, así como algunas de sus propiedades más importantes (sobre todo las que se separan de los valores por defecto). Los ficheros de este proyecto se llamarán ***colores0.vbp*** y ***colores0.frm***.

| Control | Propieda | Valor | Control | Propiedad | Valor |
|---|---|---|---|---|---|
| Frmcolores | Name | Frmcolores | Optverde | Name | Optverde |
| | Caption | Colores | | Caption | Verde |
| Fracolores | Name | Fracolor | Fraposicio | Name | Fraposicion |
| | Caption | Colores | | Caption | Posición |
| Optazul | Name | Optazul | Optarriba | Name | Optarriba |

| | Caption | Azul | | Caption | Arriba |
|---|---|---|---|---|---|
| Optrojo | Name | Optrojo | Optabajo | Name | Optabajo |
| | Caption | Rojo | | Caption | Abajo |
| Optamarillo | Name | Optamarillo | Txtcaja | Name | Txtcaja |
| | Caption | Amarillo | | Text | "" |

Tabla 1.2. Objetos y propiedades del ejemplo ***colores0***.

A continuación se muestra el código correspondiente a los procedimientos de este ejemplo.

```
Option explicit private
sub form_load()

    Txtcaja.top = 0
end sub

Private sub optarriba_click()

    Txtcaja.top = 0
end sub

Private sub optabajo_click()

    Txtcaja.top = frmcolores0.scaleheight -
txtcaja.height end sub

Private sub optazul_click()

    Txtcaja.backcolor = vbblue
end sub

Private sub optrojo_click()

    Txtcaja.backcolor = vbred
end sub

Private sub optverde_click()

    Txtcaja.backcolor = vbgreen
end sub

Private sub optamarillo_click()

    Txtcaja.backcolor = vbyellow
end sub
```

Sobre este primer programa en ***visual Basic 6.0*** se pueden hacer algunos comentarios:

1. El comando **option explicit** sirve para obligar **a *declarar*** todas las variables que se utilicen. Esto impide el cometer errores en los nombres de las variables (confundir *masa* con *mesa*, por ejemplo). En este ejemplo esto no tiene ninguna importancia, pero es conveniente acostumbrarse a incluir esta opción. **Declarar una variable** es crearla con un nombre y de un tipo determinado antes de utilizarla.
2. Cada una de las partes de código que empieza con un **private sub** y termina con un **end sub** es un **procedimiento**, esto es, una parte de código independiente y

reutilizable. El nombre de uno de estos procedimientos, por ejemplo *optazul_click()*, es típico de visual Basic. La primera parte es el nombre de un objeto (control); después va un separador que es el carácter de subrayado (_); a continuación el nombre de un evento -**click**, en este caso-, y finalmente unos paréntesis entre los que irían los argumentos, en caso de que los hubiera.

El siguiente ejemplo es utilizando la herramienta hscrollbar

## 22.COLORES RGB

La figura muestra el formulario y los controles del proyecto *colores*. Los ficheros de este proyecto se pueden llamar *colores.vbp* y *colores.frm*.

En este ejemplo se dispone de tres barras de desplazamiento con las que pueden controlarse las componentes RGB del color del fondo y del color del texto de un control ***label***. Dos botones de opción permiten determinar si los valores de las barras se aplican al fondo o al texto. Cuando se cambia del texto al fondo o viceversa los valores de las barras de desplazamiento (y la posición de los cursores) cambian de modo acorde.

Tres cajas de texto contienen los valores numéricos de los tres colores (entre 0 y 255). A la izda. Tres ***labels*** indican los colores de las tres barras.

En una aplicación nunca se debe utilizar un array o control de matrices pero en este caso si todo lo que quede programado dentro de una matriz programara automáticamente el otro control que allí pertenece

| Control | Propieda | Valor | Control | Propiedad | Valor |
|---|---|---|---|---|---|
| Frmcolore | Name | Frmcolores | Hsbcolor | Name | Hsbcolor |
| | Caption | Colores | | Min | 0 |
| Lblcuadro | Name | Lblcuadro | | Max | 255 |

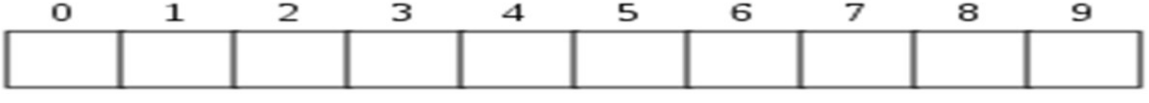
| | | | | | |
|---|---|---|---|---|---|
| | Caption | Informática 1 | | Smallchang | 1 |
| | Font | Ms sans serif, bold, 24 | | Largechange | 16 |
| Cmdsalir | Name | Cmdsalir | | Index | 0,1,2 |
| | Caption | Salir | | Value | 0 |
| | Font | Ms sans serif, bold, 10 | Txtcolor | Name | Txtcolor |
| Optcolor | Name | Optcolor | | Text | 0 |
| | Index | 0,1 | | Locked | True |
| | Caption | Fondo, texto | | Index | 0,1,2 |
| | Font | Ms sans serif, bold, 10 | Lblcolor | Name | Lblcolor |
| | | | | Caption | Rojo,verde,azul |
| | | | | Index | 0,1,2 |
| | | | | Font | Ms sans serif, 10 |

Objetos y propiedades del ejemplo ***colores***.

Una característica importante de este ejemplo es que se han utilizado **vectores (arrays) de controles o control de matrices**. Las tres barras se llaman **hsbcolor** y se diferencian por la propiedad **index**, que toma los valores 0, 1 y 2. También las tres cajas de texto, las tres **labels** y los dos botones de opción son ***arrays de controles.*** Para crear un array de controles basta crear el primero de ellos y luego **copiar** y **pegar** tantas veces como se desee, respondiendo afirmativamente al cuadro de diálogo que pregunta si desea crear un array.

**El *procedimiento sub*** que contiene el código que gestiona un **evento** de un array es único para todo el array, y recibe como argumento la propiedad ***index***. De este modo que se puede saber exactamente en qué control del array se ha producido el evento. Así pues, una ventaja de los *arrays* de controles es que pueden compartir el código de los eventos y permitir un tratamiento conjunto

**Un array:** es un medio de guardar un conjunto de objetos de la misma clase. Se accede a cada elemento individual del array mediante un número entero denominado índice. 0 es el índice del primer elemento y *n-1* es el índice del último elemento, siendo *n*, la dimensión del array.




En cada pedazo de memoria dentro de un arreglo o array con su respectivo índice se almacena un valor determinado

Por medio de bucles ***for.*** A continuación se muestra el código correspondiente a los procedimientos que tratan los eventos de este ejemplo.

por medio de bucles *for*. A continuación se muestra el código correspondiente a los procedimientos que tratan los eventos de este ejemplo.

```
Option Explicit
Public Brojo, Bverde, Bazul As Integer
Public Frojo, Fverde, Fazul As Integer

Private Sub cmdSalir_Click()
   End
End Sub

Private Sub Form_Load()
   Brojo = 0
   Bverde = 0
   Bazul = 0
   Frojo = 255
   Fverde = 255
   Fazul = 255
   lblCuadro.BackColor = RGB(Brojo, Bverde, Bazul)
   lblCuadro.ForeColor = RGB(Frojo, Fverde, Fazul)
End Sub

Private Sub hsbColor_Change(Index As Integer)
   If optColor(0).Value = True Then
      lblCuadro.BackColor = RGB(hsbColor(0).Value, hsbColor(1).Value, _
                            hsbColor(2).Value)
      Dim i As Integer
      For i = 0 To 2
         txtColor(i).Text = hsbColor(i).Value
      Next i
   Else
      lblCuadro.ForeColor = RGB(hsbColor(0).Value, hsbColor(1).Value, _
                            hsbColor(2).Value)
      For i = 0 To 2
         txtColor(i).Text = hsbColor(i).Value
      Next i
   End If
End Sub

Private Sub optColor_Click(Index As Integer)
   If Index = 0 Then  'Se pasa a cambiar el fondo
      Frojo = hsbColor(0).Value
      Fverde = hsbColor(1).Value
      Fazul = hsbColor(2).Value
      hsbColor(0).Value = Brojo
      hsbColor(1).Value = Bverde
      hsbColor(2).Value = Bazul
   Else                'Se pasa a cambiar el texto
      Brojo = hsbColor(0).Value
      Bverde = hsbColor(1).Value
      Bazul = hsbColor(2).Value
      hsbColor(0).Value = Frojo
      hsbColor(1).Value = Fverde
      hsbColor(2).Value = Fazul
   End If
End Sub
```

**For** de este ejercicio significa por ejemplo for i =0 to 2 significa que empiece desde o hasta 2 y cuando llegue a dos pare.

El código de este ejemplo es un poco más complicado que el de los ejemplos anteriores y requiere unas ciertas explicaciones adicionales adelantando cuestiones que se verán posteriormente:

1. La función **RGB()**crea un **código de color** a partir de sus argumentos: las componentes RGB(*red*, *green* and *blue*). Estas componentes, cuyo valor se

almacena en un byte y puede oscilar entre 0 y 255, se determinan por medio de las tres barras de desplazamiento.

2. El color **blanco** se obtiene con los tres colores fundamentales a su máxima intensidad. El color **negro** se obtiene con los tres colores RGB a cero. También se pueden introducir con las constantes predefinidas **vbwhite** y **vbblack**, respectivamente.
3. Es importante disponer de unas **variables globales** que almacenen los colores del fondo y del texto, y que permitan tanto guardar los valores anteriores de las barras como cambiar éstas a sus nuevos valores cuando se clica en los botones de opción. Las variables globales, definidas en la parte de definiciones generales del código, fuera de cualquier procedimiento, son visibles desde cualquier parte del programa. Las variables definidas dentro de una función o procedimiento sólo son visibles desde dentro de dicha función o procedimiento (*variables locales*).
4. La función **hsbcolor change(index as integer)**se activa cada vez que se cambia el valor en una cualquiera de las barras de desplazamiento. El argumento ***index***, que **visual BASIC** define automáticamente, indica cuál de las barras del array es la que ha cambiado de valor (la 0, la 1 ó la 2). En este ejemplo dicho argumento no se ha utilizado, pero está disponible por si se hubiera querido utilizar en el código.

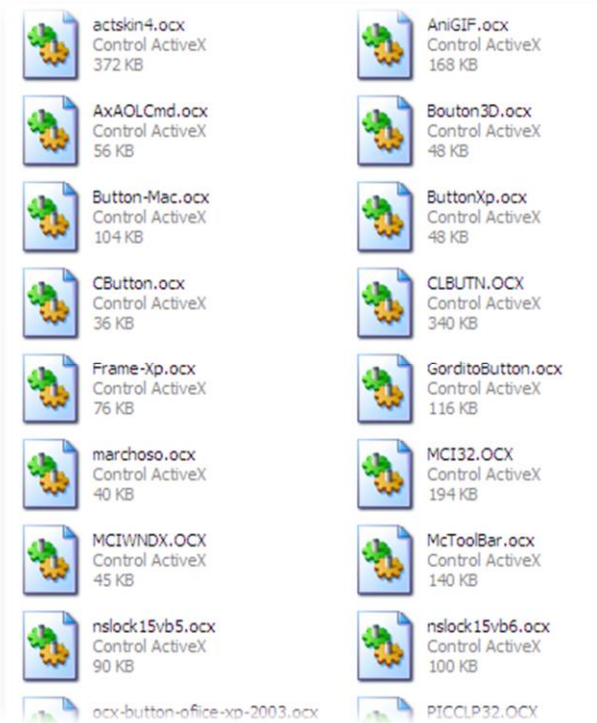
Hacer el ejercicio anterior del ejemplo nuevamente

# 23. USO DE OCX ALTERNAS DE VISUAL BASIC

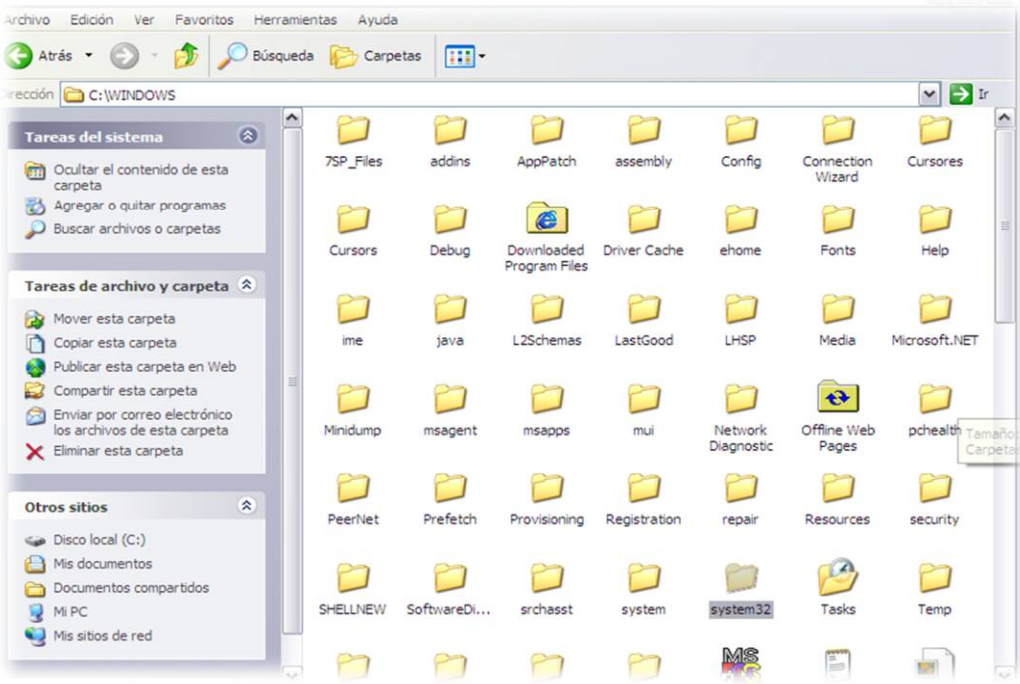
En este ejemplo se va a utilizar otros controles ocx de visual Basic a continuación haremos un ejemplo utilizando multimedia para una rokola

**Ocx:** hace referencia a instrucciones preprogramadas capaces de ser insertadas en cualquier aplicación que está siendo desarrollada, y permite al programador ahorrar tiempo debido a que en lugar de reprogramar un código anteriormente hecho, ahora únicamente lo inserta.

Las funciones de una ocx son ilimitadas, únicamente depende de que cualquier programador la desarrolle e implemente para futuros usos.

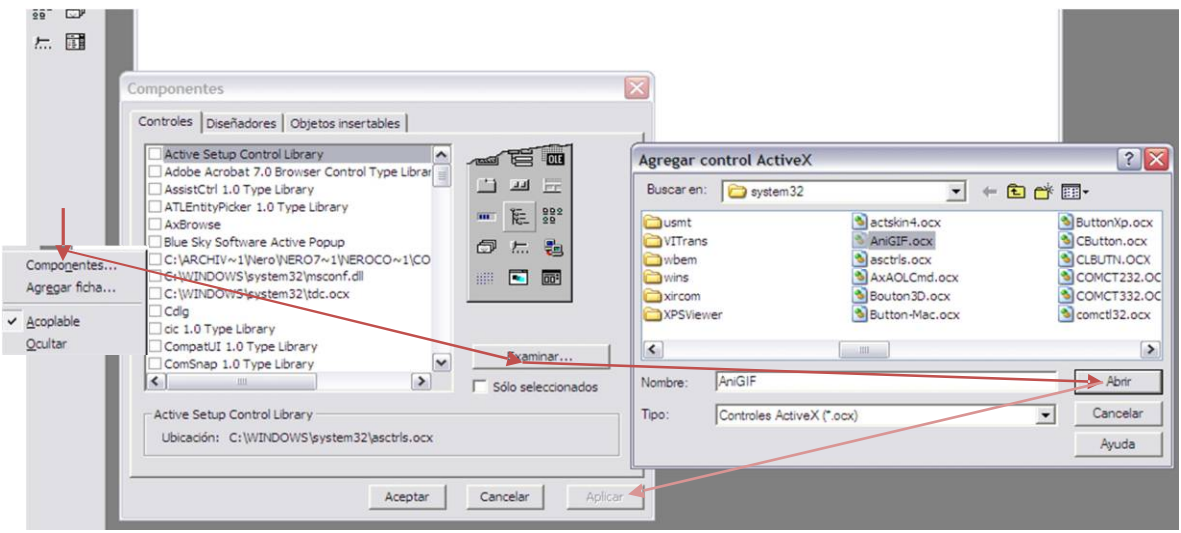



Copiamos todo estos controles a **mi pc, disco local c, carpeta Windows, carpeta system 32** si no tenemos estas ocx se pueden descargar de internet buscando en google ocx para visual Basic 6.0




Después de pegarlos los cargamos a visual de la siguiente manera

Damos click derecho sobre la barra de controles luego componentes luego click en el botón examinar elegimos el control y abrir luego aplicar y aceptar



Controles a utilizar en este ejemplo

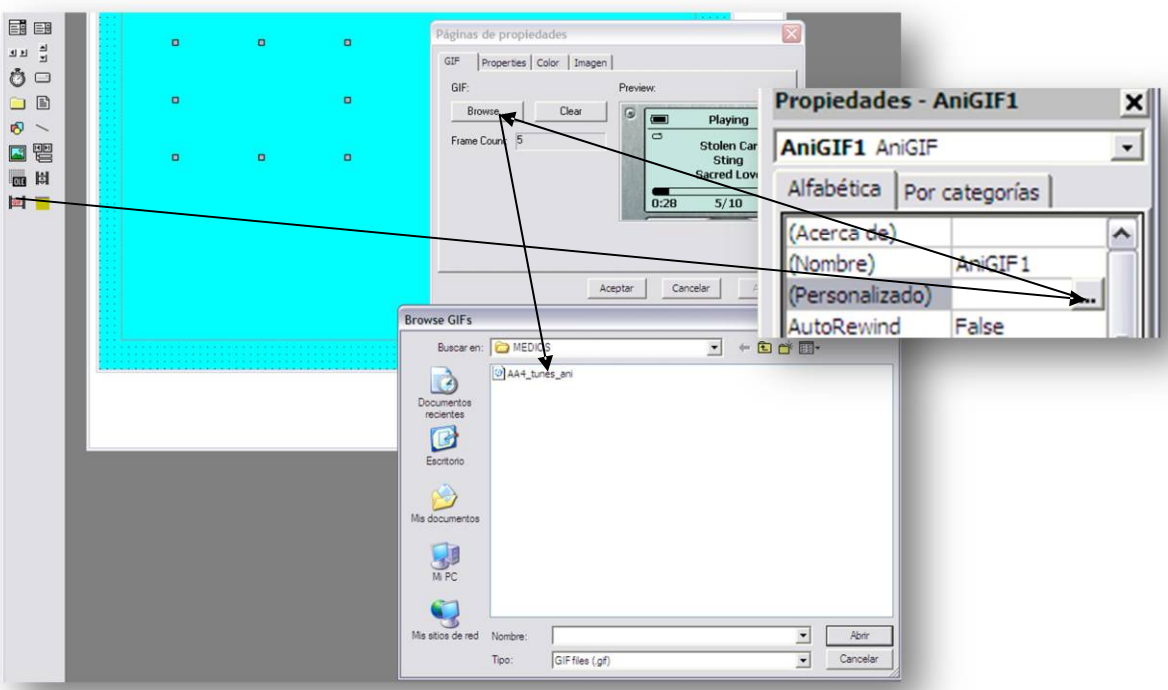
**Anigif.ocx=**sirve para cargar imágenes de tipo animadas

**Mciwndx.ocx=**se utiliza para cargar la música al proyecto de visual

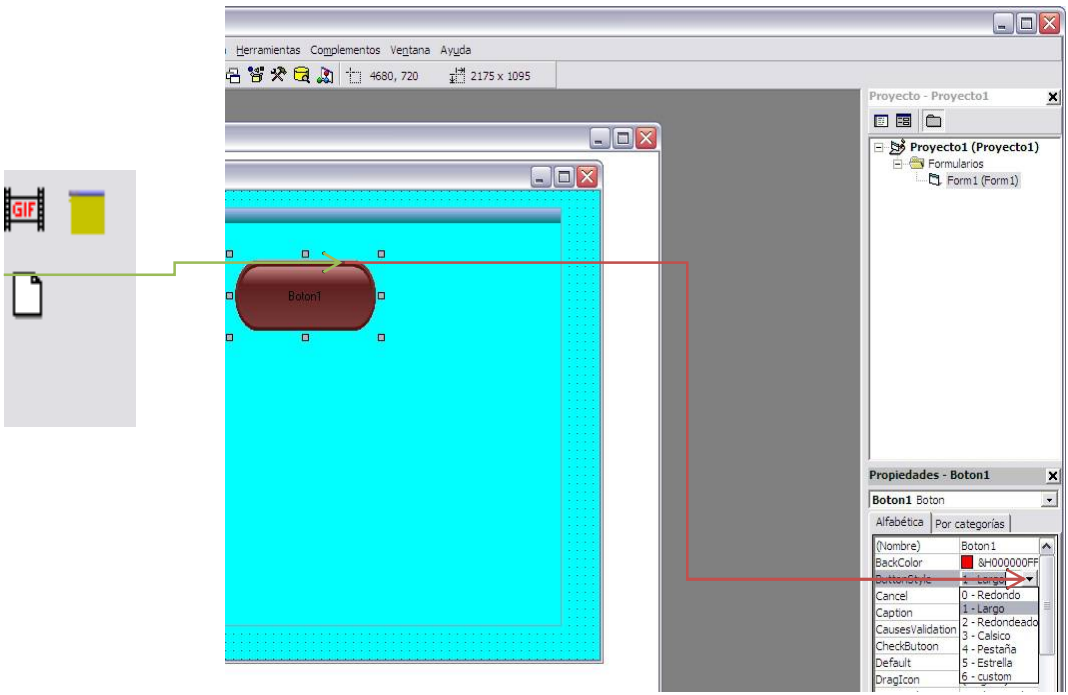
**Gorditobutton.ocx=** se utiliza para cargar otros botones pero animados al proyecto

**Framexp.ocx=**se utiliza como fondo del formulario para contener otros controles

Recuerden que las imágenes y el audio deben estar en la carpeta formas dentro de la carpeta principal del software ya creada posteriormente en archivos de programa como ya se explicó anteriormente en esta guía para cargar el anigif el control dibujamos el anigif y luego en propiedades tal como muestra en la gráfica Luego en el botón browse cargamos la imagen y listo



El anigif debe estar **visible false** en las propiedades se activa con los botones a continuación veremos cómo utilizar los gorditon button se dibuja el **gorditonbutton** tal como aparece en la imagen él tiene varios estilos tal como muestra en la barra de propiedades

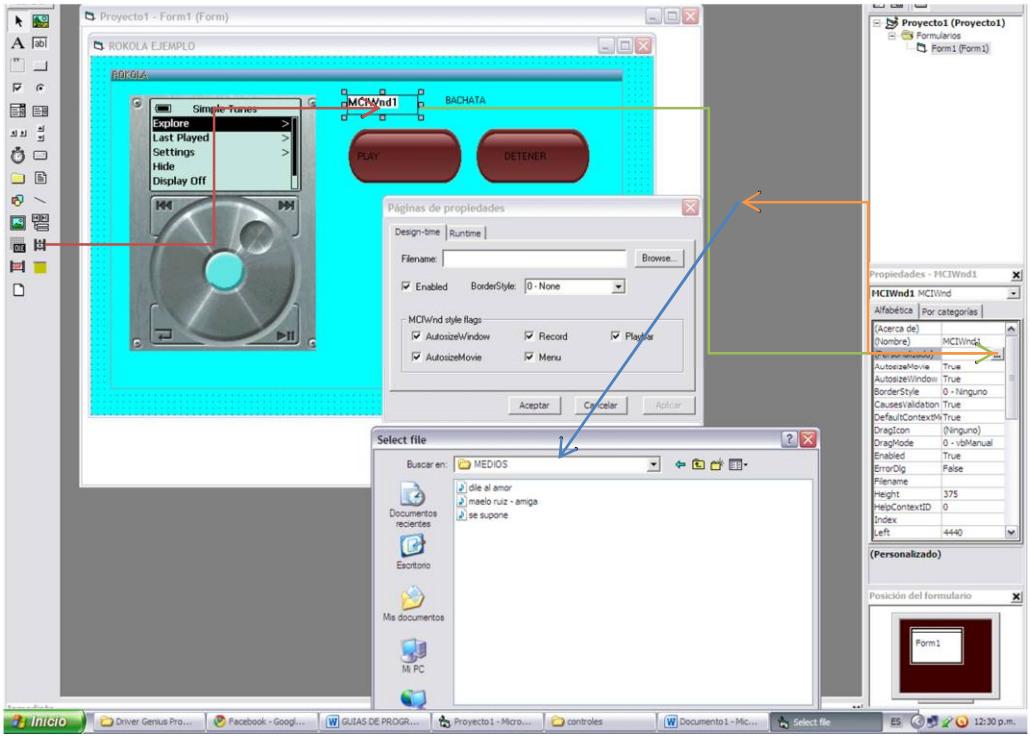



Para que el botón quede animado modifico las siguientes propiedades

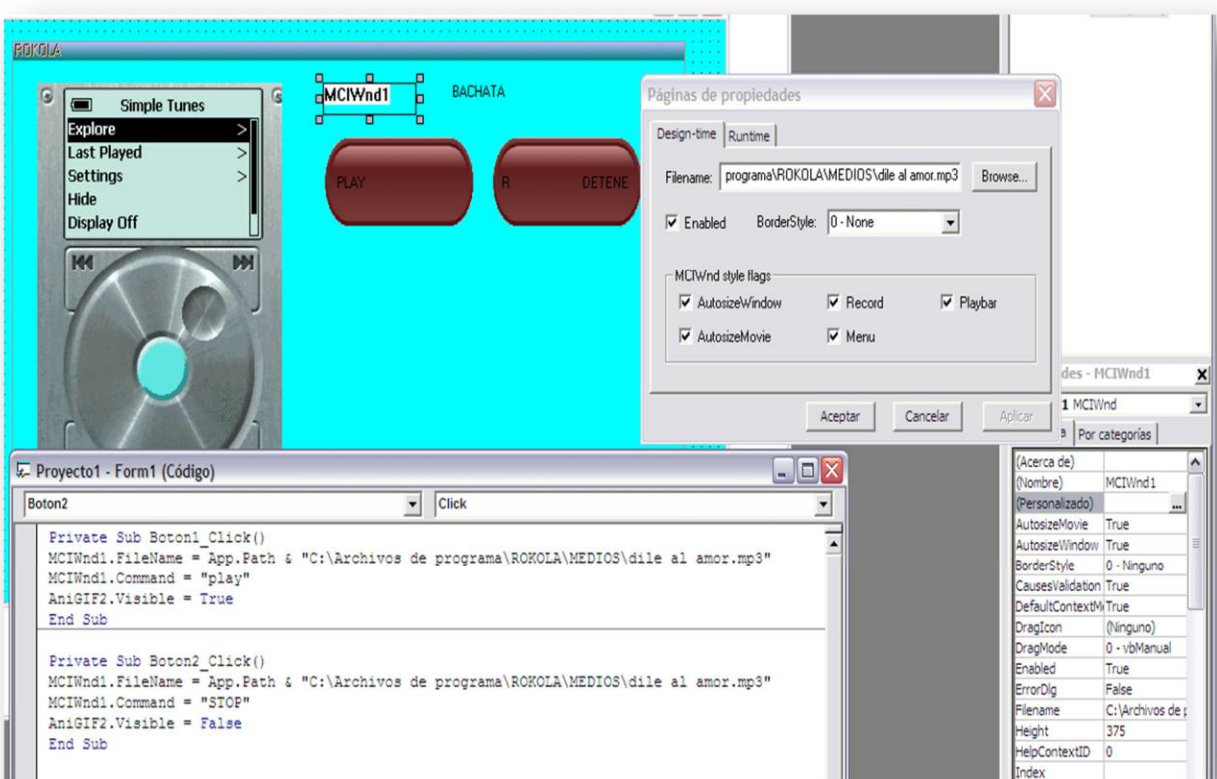
**Caption** se cambia el nombre

**Usemarquesina**=true y ya

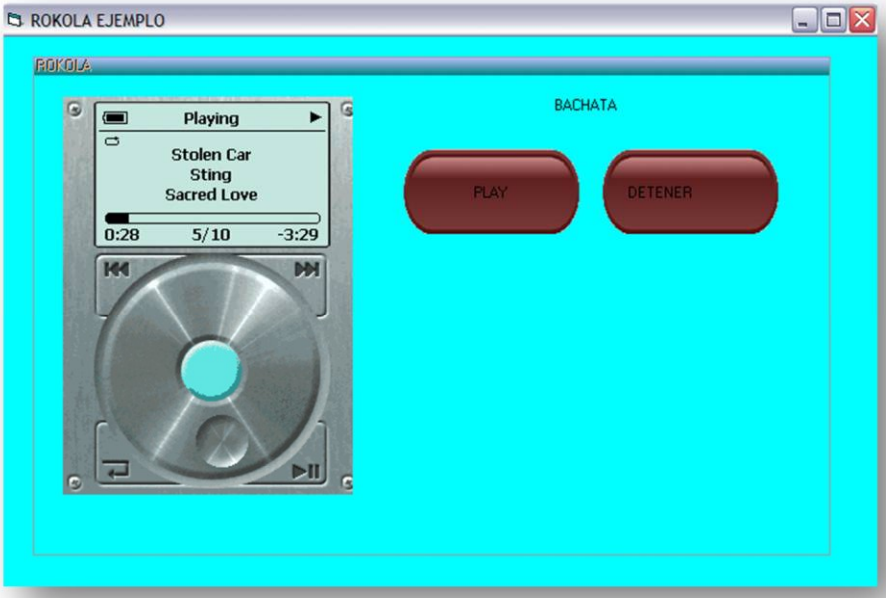
Para cargar el audio se hace de la siguiente manera recordemos que solo podemos cargar audio con formato **mp3.wap.y midi**




El Control De Sonido De Estar Visible False La Programación Ya En Si De Esta Mini Aplicación Queda Así



**filename** es la ruta del archivo tal como vemos en la gráfica en ejecución queda así




Ejercicio de la guía terminar la rokola con más canciones y cosas

# 24. INSTRUCCIONES DE REPETICION

**Do while** condición el do while significa mientras que haga realiza la instrucción

Instruccion1

Instruccion2

**Loop** significa repetir varias veces un número determinado de instrucciones a continuación un ejemplo de esta instrucción con factorial doy una breve explicación de factorial para que entienda la instrucción que voy a programar

## Factorial

La **función factorial** (símbolo: **!**) Sólo quiere decir que se multiplican una serie de números que descienden. Ejemplos:

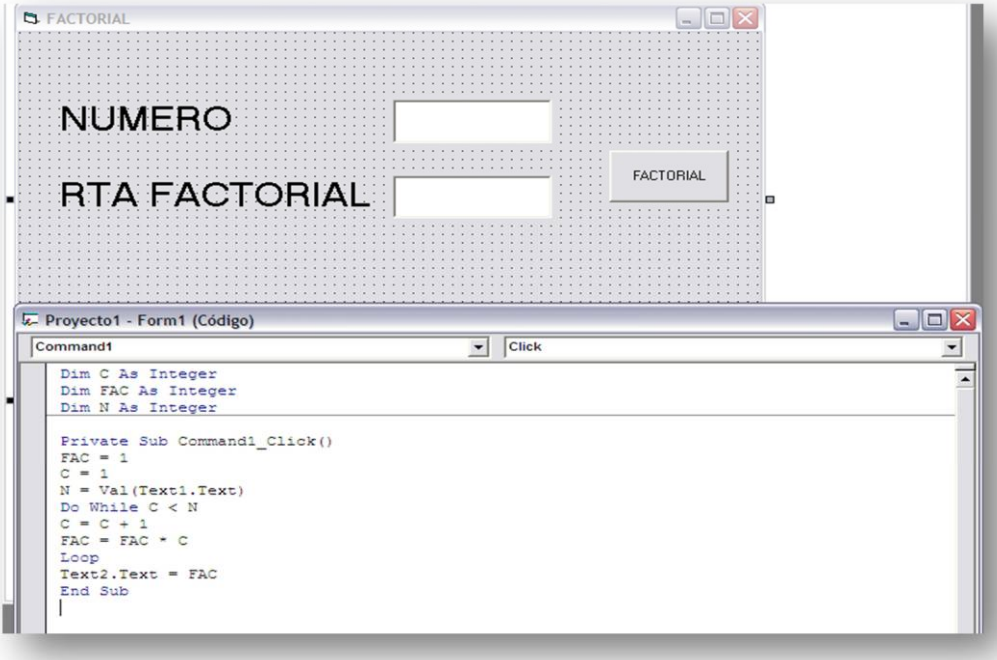
- $4! = 4 \times 3 \times 2 \times 1 = 24$
- $7! = 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 5040$
- $1! = 1$

"4!" normalmente se pronuncia "4 factorial". También se puede decir "factorial de 4"

Calculando desde el valor anterior

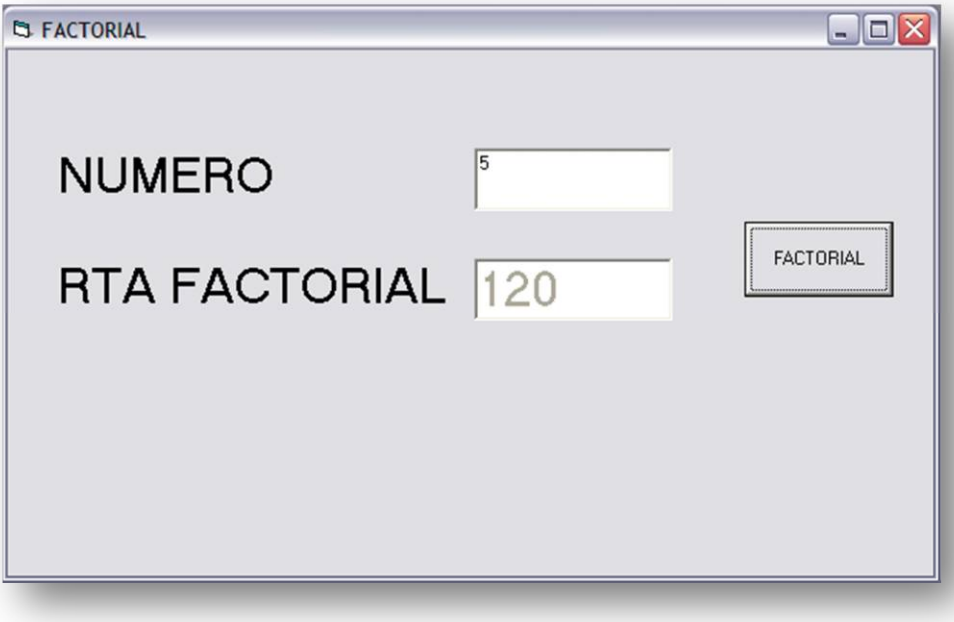
Es fácil calcular un factorial desde el valor anterior:

| N | N! | | |
|---|---|---|---|
| 1 | **1** | 1 | 1 |
| 2 | $2 \times$ **1** | $= 2 \times$ **1!** | = 2 |
| 3 | $3 \times$ **2** $\times$ **1** | $= 3 \times$ **2!** | = 6 |
| 4 | $4 \times$ **3** $\times$ **2** $\times$ **1** | $= 4 \times$ **3!** | = 24 |
| 5 | $5 \times$ **4** $\times$ **3** $\times$ **2** $\times$ **1** | $= 5 \times$ **4!** | = 120 |
| 6 | Etc. | Etc. | |




Recuerden que la caja de resultados va **enabled false** para que no puedan modificar su contenido enabled activar y desactivar el control

En ejecución queda así



# 25. BASE DE DATOS ACCES

## Sistema de base de datos

Este sistema permite almacenar, manipular, y recuperar información,

1. **Base de datos:** es un conjunto de archivos interrelacionados y almacenados electrónicamente.

2. **Data base manager (d.b.m)**

El administrador de la base de datos es la persona responsable del diseño físico de la base de datos además se encarga de evaluar, seleccionar e implementar software administrador de la base de datos.

3. **Software administrador de la base de datos data base manogemetsistems)**

Es el software que controla, organiza, almacena y recupera, se encarga de la seguridad de la información, acepta, solicitudes de una aplicación y ordena al sistema operativo transferir datos apropiados.

4. **Hardware**

La base de datos puede almacenarse en varias máquinas de igual o diferente tipo.

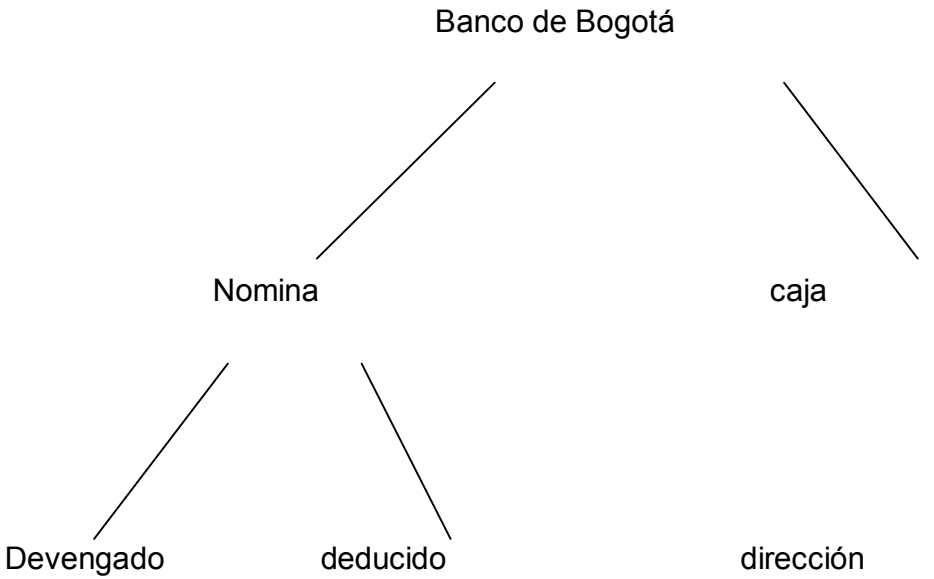
5. **Usuarios**

Programadores, digitadores, administradores, usuarios finales

¿Cómo está la información en una base de datos?

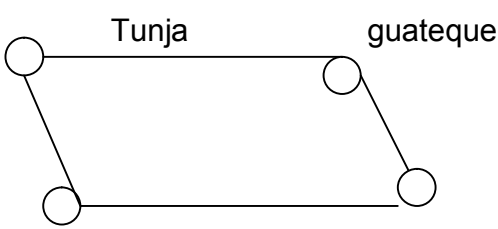
En una o varias bases de datos

Manejo base de datos




Que los datos procedentes a la misma base de datos pueda almacenarse en sitios diferentes.

Cuentas de ahorros




Bogotá

Medellín

## Información integrada

Elimina la redundancia, cuando la información está integrada se puede tener en una o varias bases de datos informes de diferentes asuntos y eliminando total o parcialmente la redundancia.

Información compartida la usan diferentes usuarios para diferentes propósitos

## Partes de una base de datos

## Entidad

Es un objeto relevante o importante para determinado entorno de la cual podemos representar información.

## Ejemplo

Estudiante una regla importante es que va en mayúscula y en singular

## Atributos

Son las propiedades o características de las entidades

## Ejemplo

Estudiante

Cc

Nombre

Apellido

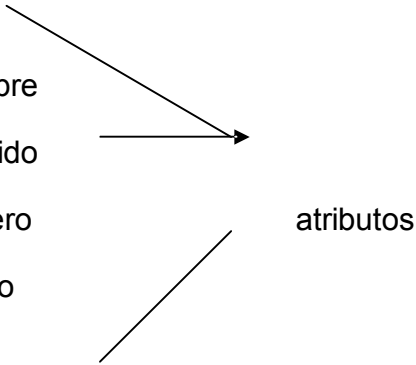
Genero

Grado

Edad

Dirección

Teléfono

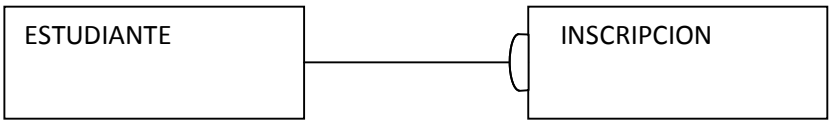


## Relaciones

Ir en un cuadrado redondeado a los bordes

Se representa con una línea recta

1 a 1

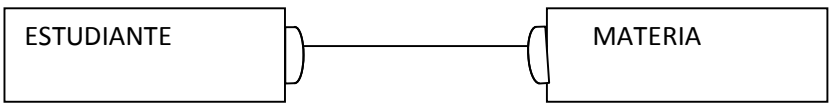



Un estudiante solo realiza una inscripción.

Uno a muchos

Un hombre árabe tiene varias mujeres

Muchos a muchos

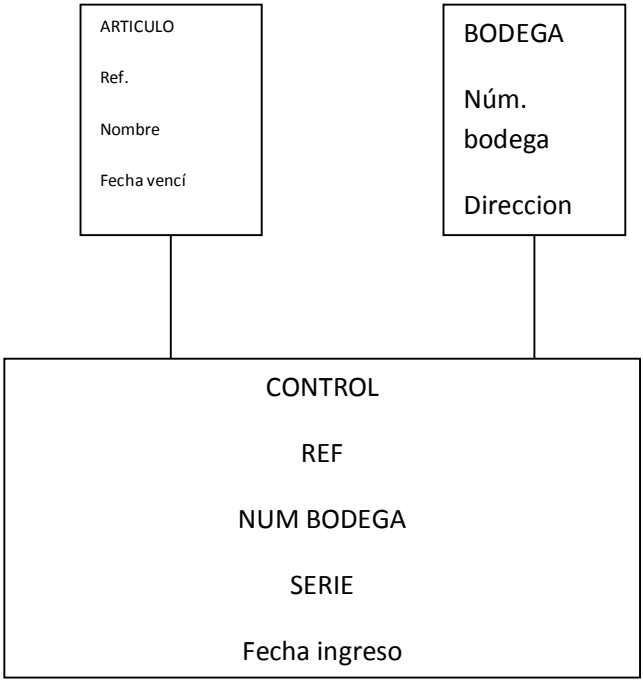



Varios estudiantes ven varias materias y varias materias son vistas por varios estudiantes.

**Identificación: conocida como llave primaria o clave principal** permite distinguir de manera única una entidad puede tener uno o varios atributos.

## Rompimiento

Un rompimiento nace de una destrucción de una relación muchos a muchos surge una tercera entidad la cual arrastra las llaves primarias de las entidades anteriores a quien dirige dos relaciones de 1 a muchos se simboliza con una línea sobre el símbolo muchos.



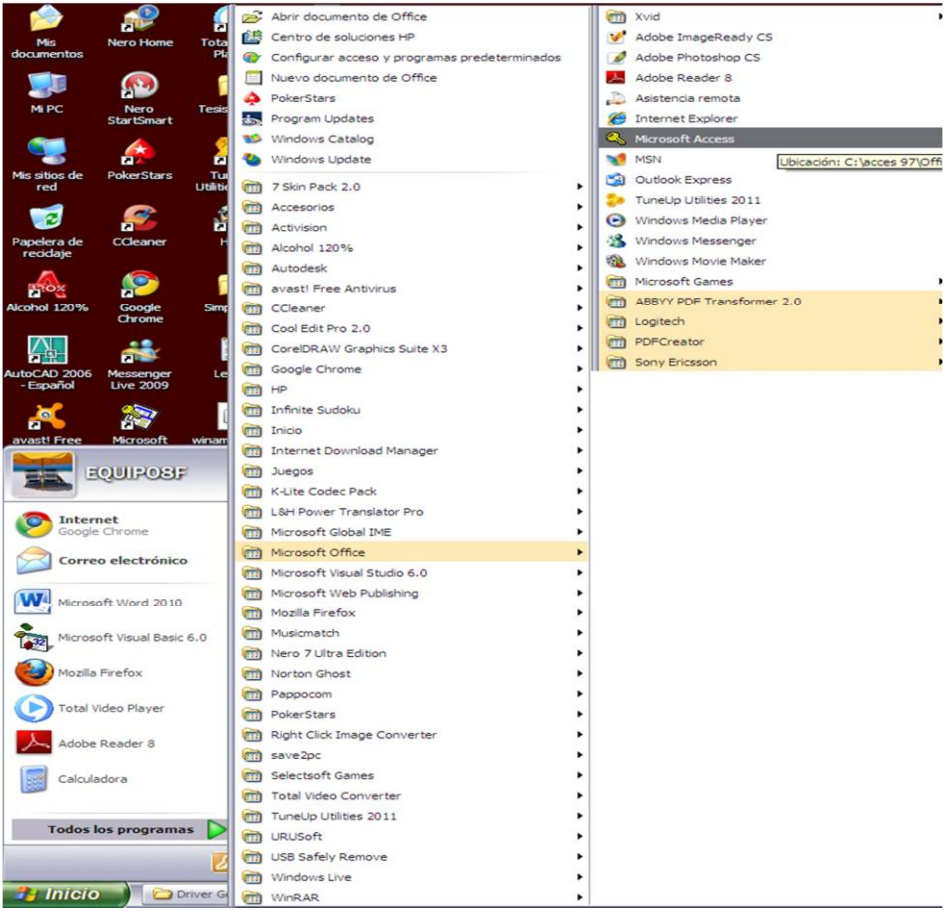

## Modelo entidad relación

Diseño global de la base de datos

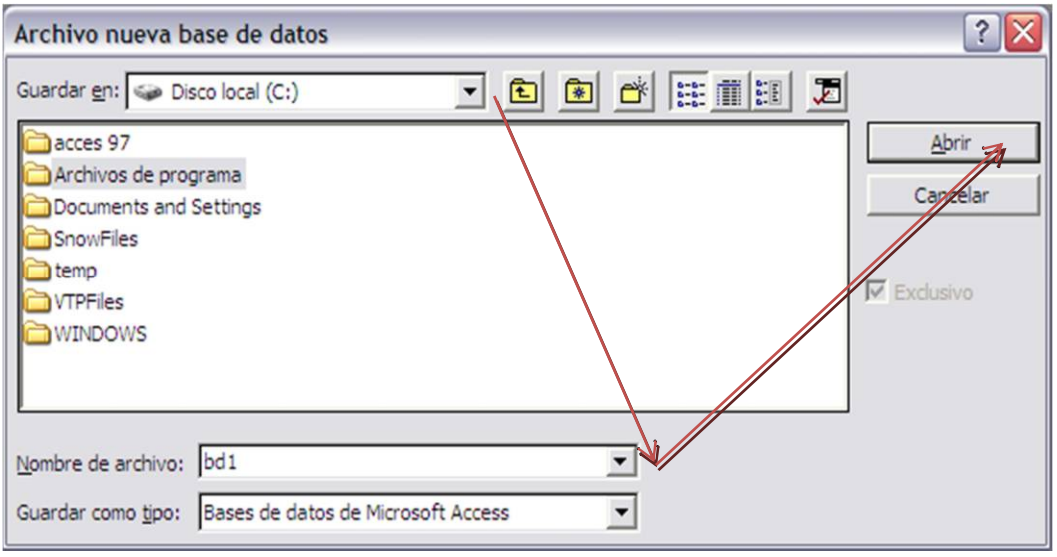
1. Mayúscula
2. Singular
3. .no relaciones de muchos a muchos
4. Derecha asía arriba e izquierda
5. No líneas cruzadas.

Para abrir una base de datos en blanco click base de datos en blanco y aceptar.

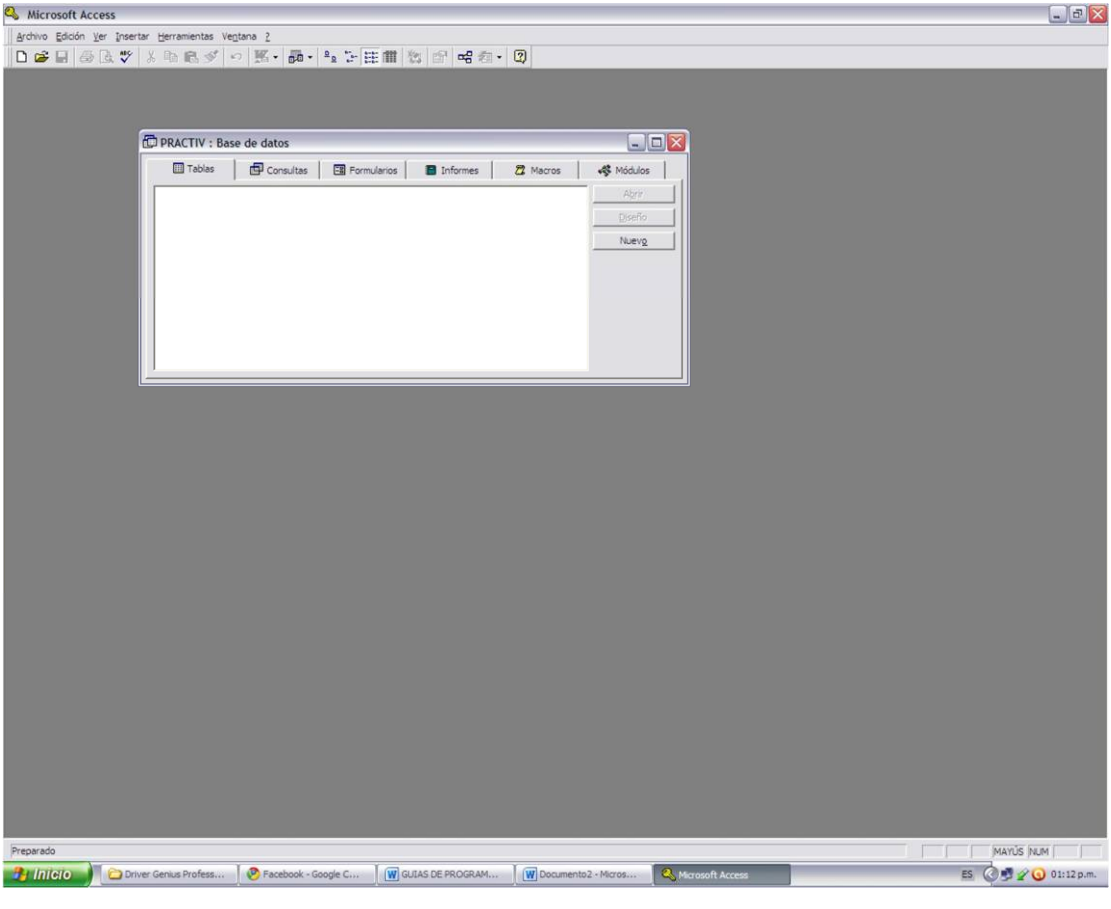
Para entrar a Access 97 tal como está en la gráfica siguiente



Click en la opción base de datos en blanco

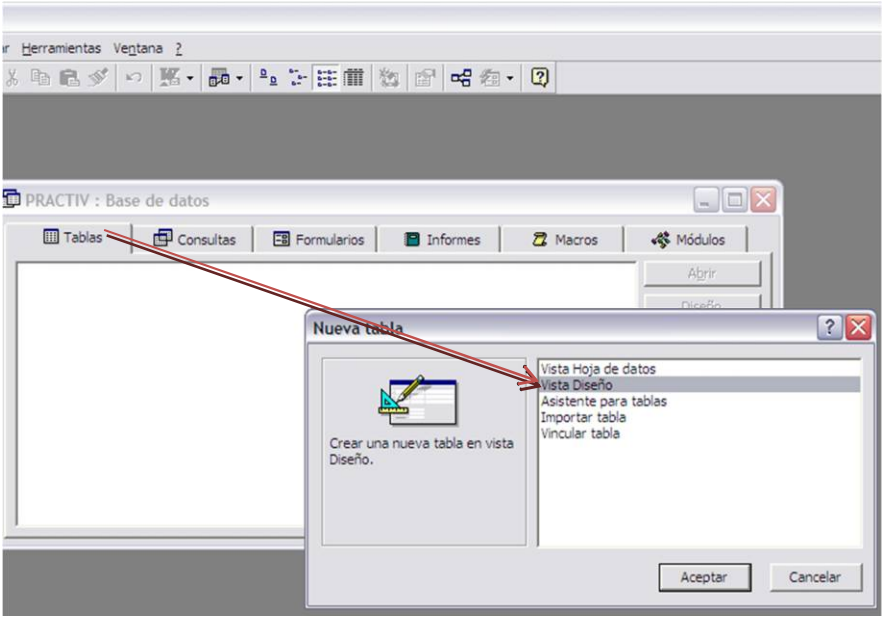


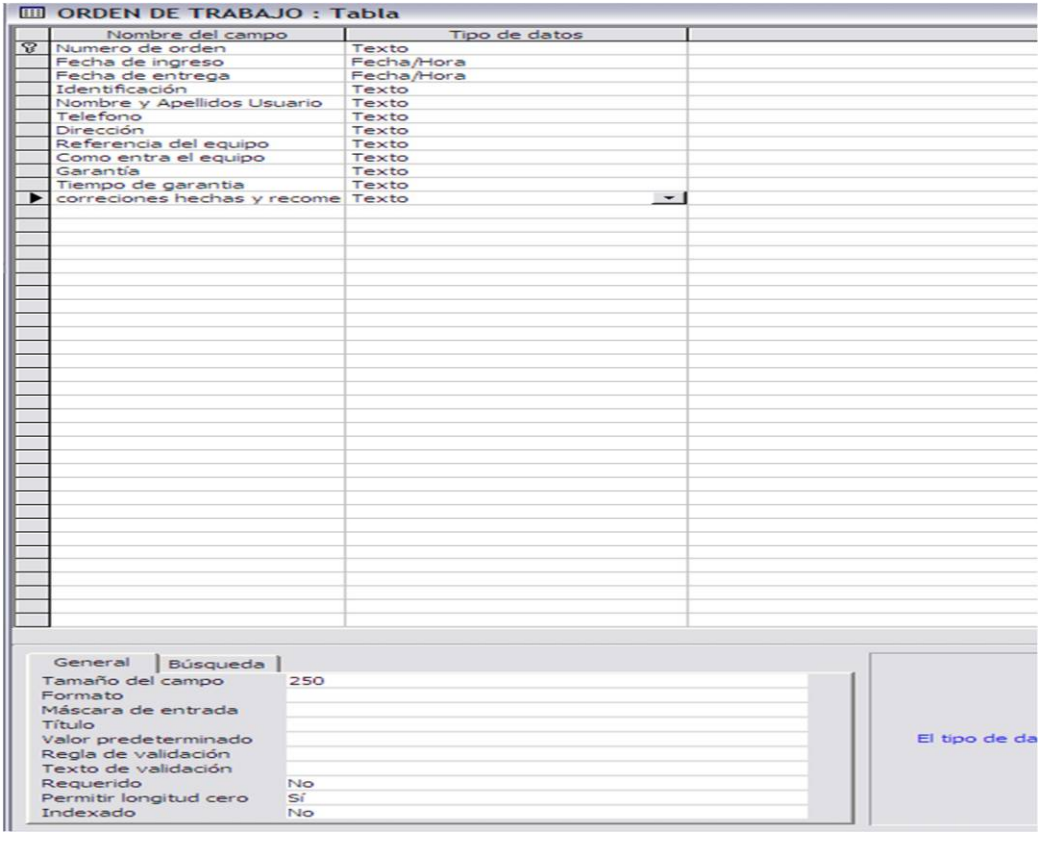

Luego doy el nombre de la base de datos elijo la ruta de guardado y abrir y luego queda así



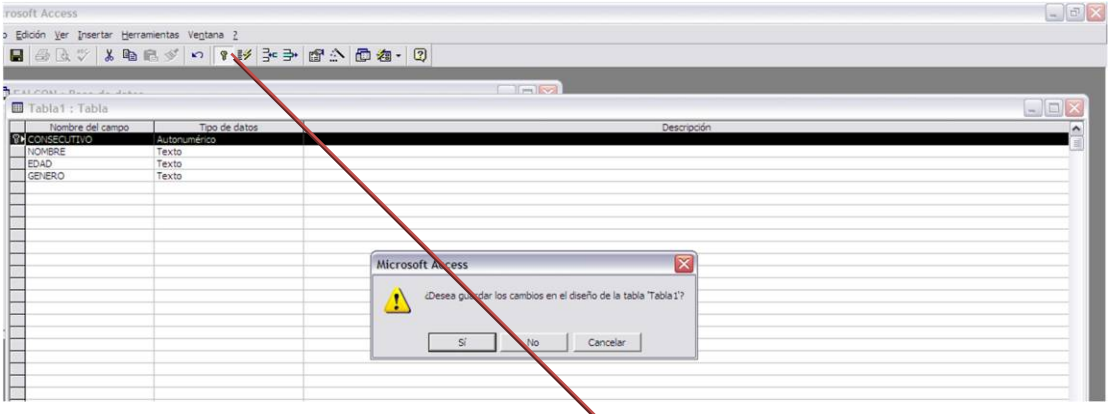
A continuación crearemos una mini base de datos

Creación de tablas o entidades click en la pestaña tabla  luego vista de diseño

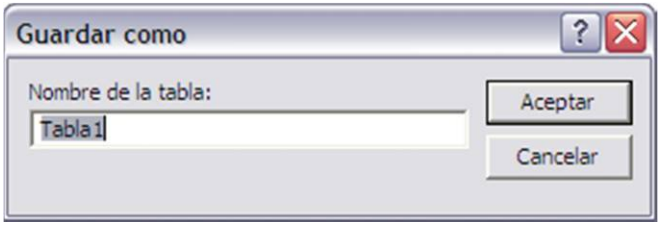


La columna **nombre del campo** es el atributo de la entidad tal como se explicó antes de este ejemplo en la guía en el campo **tipo de datos** es el tipo de información que se va a guardar en cada atributo de esta entidad **el tipo texto** es para guardar números y letras para que quepa másinformación el tamaño del campo debe ser de **250** al crear **la llave primaria** debe ser **indexado si** pero sin duplicados, para cuando queremos que en un campo nos deje guardar valores vacío o cero colocamos requerido no y permitir longitud cero si queremos que Access nos coloque un serial automático cada vez que guardemos datos en la llave primaria se deja **autonumerico** luego haremos esto

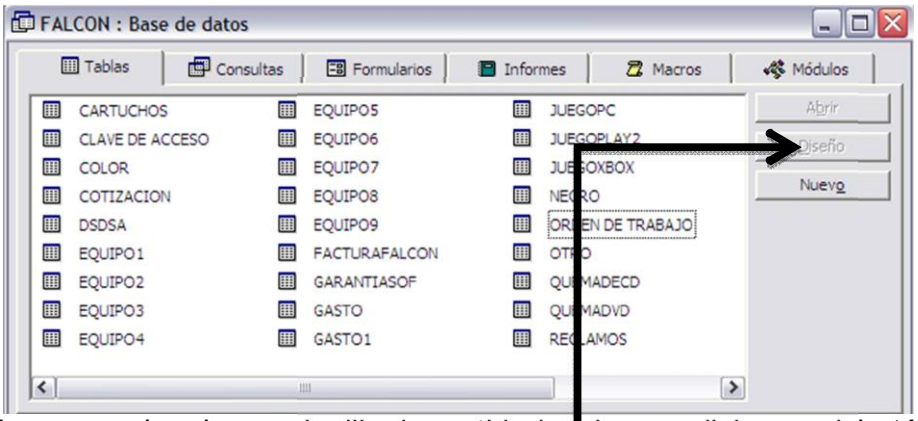



Luego click en la x roja y le digo que si deseo guardar los datos para colocar la llave primaria seleccionamos el campo que elegimos como llave primaria y luego pulsamos en la llave que está en la parte superior tal como muestra la gráfica después de hacer todo esto sale esto




Aquí le damos el nombre a la entidad recordemos que debe ser en mayúscula y en singular el nombre de la entidad

Cada que creemos una tabla o entidad en la base de datos quedara así



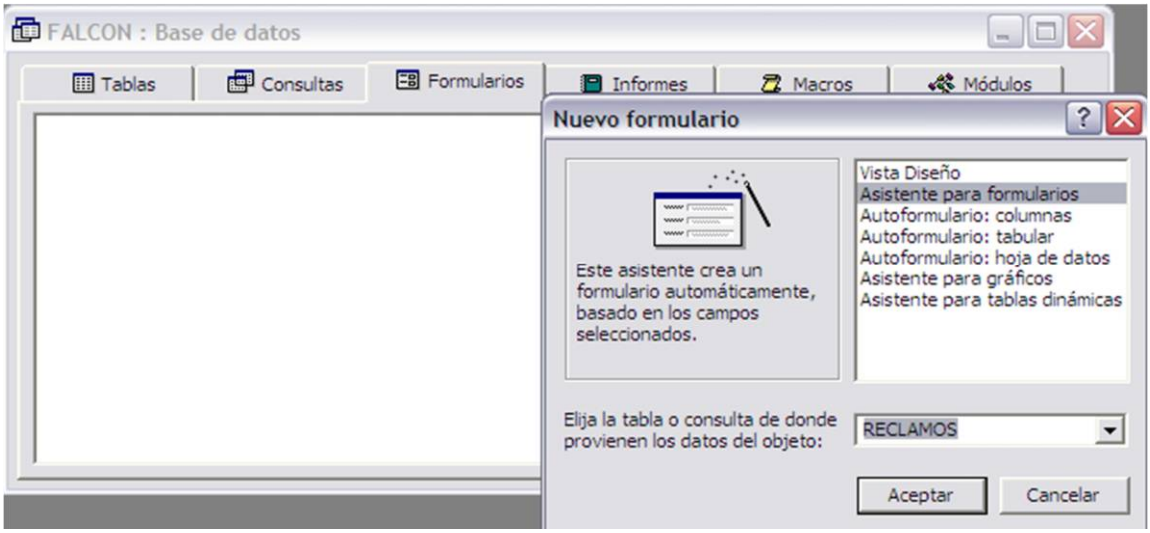

Si me quedo algo mal elijo la entidad y luego click en el botón diseño para modificar

Para guardar datos damos doble click sobre la entidad y saldrá esto

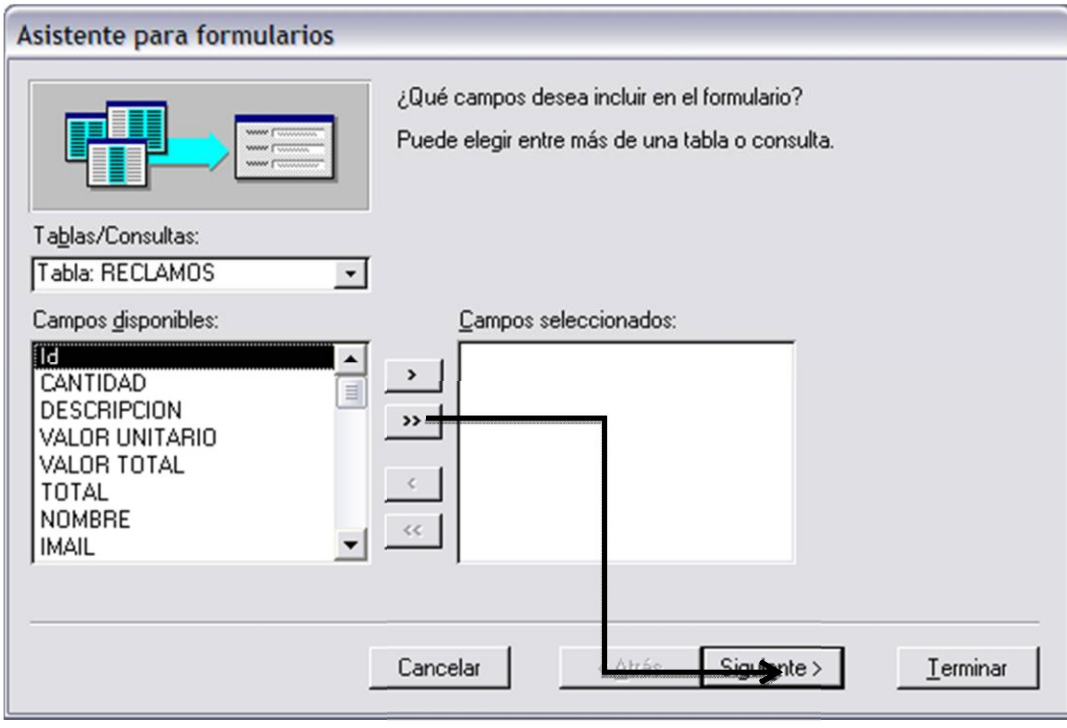
RECLAMOS : Tabla

| CAN | DESCRIPCION | VALOR UNITA | VALOR TOTA | TOTAL | NOMBRE | IMAIL | TEL | FECHA | HORA | ABONO | SALDO | referenci |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | ANAKFKJSABFKJ | 40000 | 80000 | 80000 | DAVE | LASJDHFSKJHAF | 31200000000 | 10-01-2009 | 11:35:24 AM | | | 32 |
| 3 | AKLNDKJANDFKJASBNK | 5000 | 15000 | 95000 | DAVE | LASJDHFSKJHAF | 31200000000 | 10-01-2009 | 11:35:29 AM | | | 32 |
| 4 | ASKLFJSAKLF | 50000 | 200000 | 295000 | DAVE | LASJDHFSKJHAF | 31200000000 | 10-01-2009 | 11:35:33 AM | | | 32 |
| 2 | LAKMKLADNKL | 5000 | 10000 | 305000 | DAVE | LASJDHFSKJHAF | 31200000000 | 10-01-2009 | 11:35:39 AM | | | 32 |
| 3 | KJLNDKJWEHKWJ | 3000 | 9000 | 314000 | DAVE | LASJDHFSKJHAF | 31200000000 | 10-01-2009 | 11:35:45 AM | | | 32 |
| 3 | adskjfhajfj | 20000 | 60000 | 60000 | Andres rey | sdjfgjsgfs | 887687687 | 10-01-2009 | 11:51:10 AM | | | 34 |
| 3 | fknsadkjhdfkaj | 50000 | 150000 | 210000 | Andres rey | sdjfgjsgfs | 887687687 | 10-01-2009 | 11:51:14 AM | | | 34 |
| 3 | lakjedbhkjhge | 5000 | 15000 | 225000 | Andres rey | sdjfgjsgfs | 887687687 | 10-01-2009 | 11:51:17 AM | | | 34 |
| 4 | kjshdkjahdkha | 500 | 2000 | 227000 | Andres rey | sdjfgjsgfs | 887687687 | 10-01-2009 | 11:51:25 AM | | | 34 |

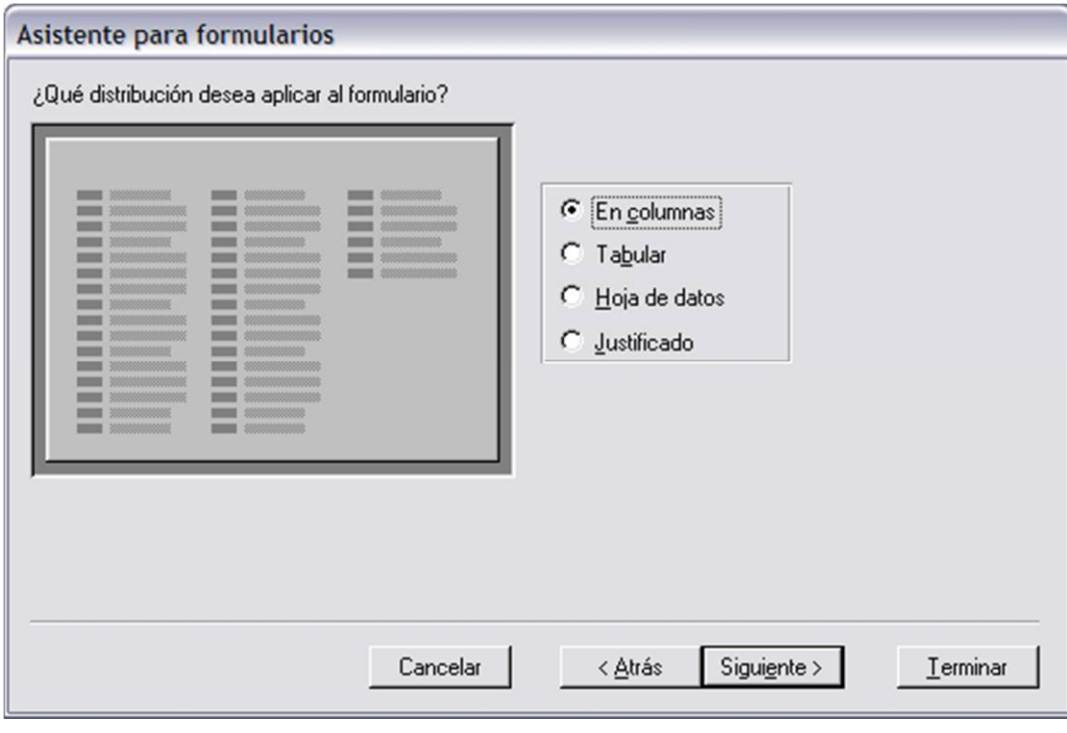
Para crear formularios de llenado de datos hacemos lo siguiente



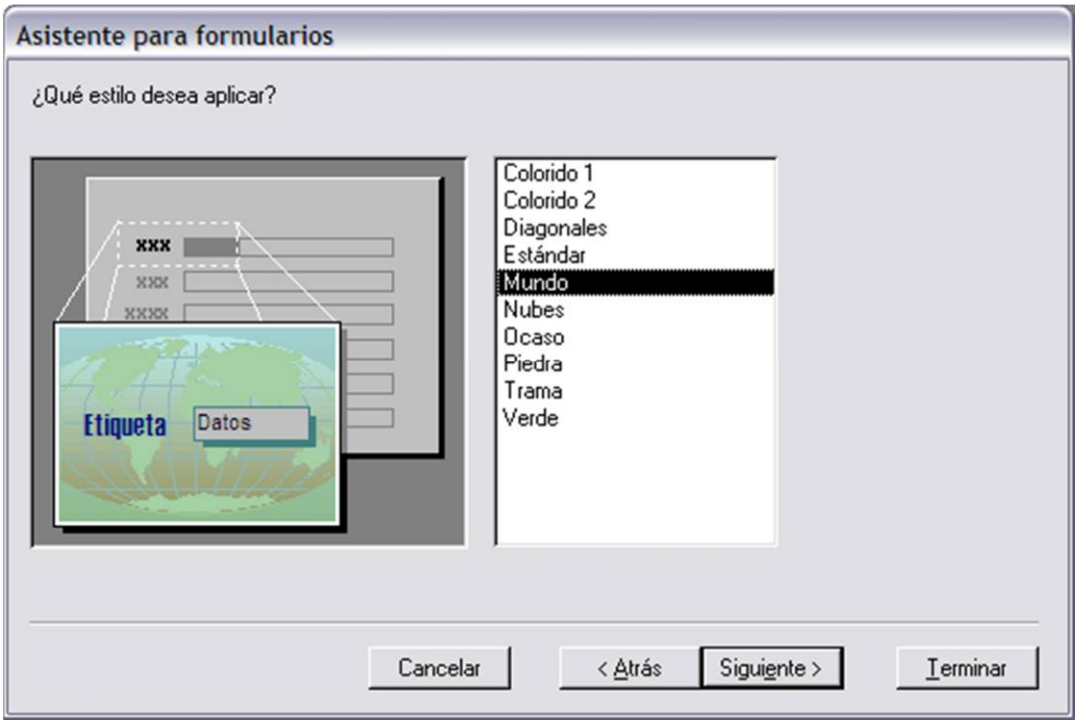

Click en la pestaña formularios asistente para formularios elijo la tabla y luego aceptar



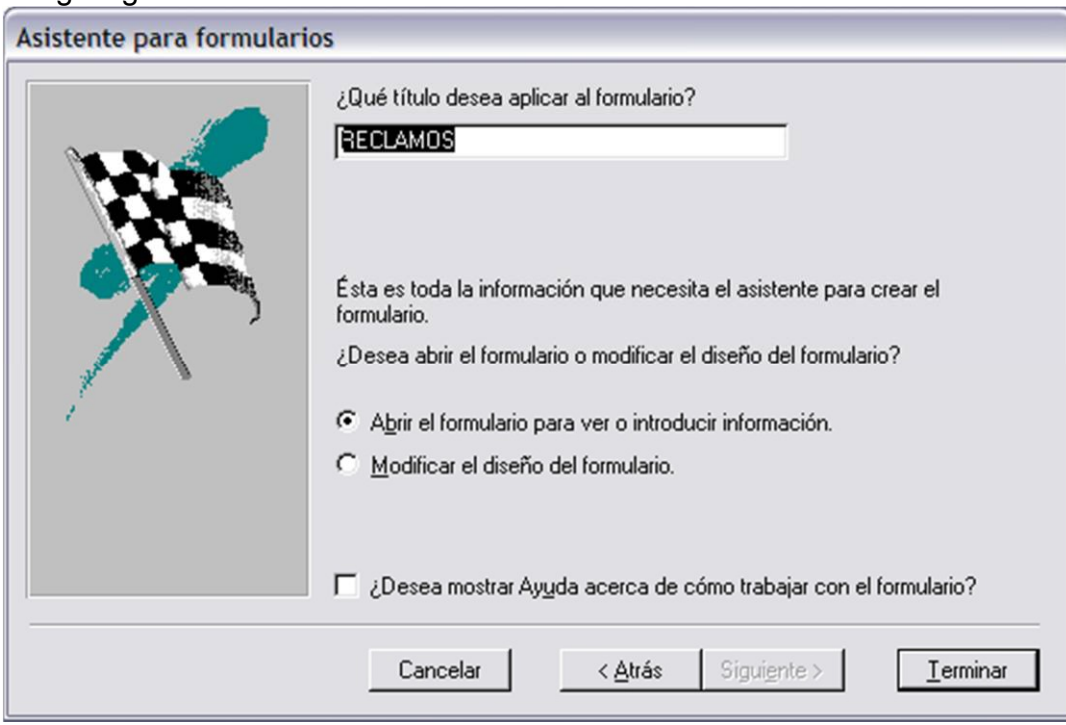

Luego agrego cada atributo de la entidad con estos botones luego siguiente



Elegir un tipo ya sea en columnas tabulado hoja de datos o justificado. En justificado queda así después de esto elegir un estilo



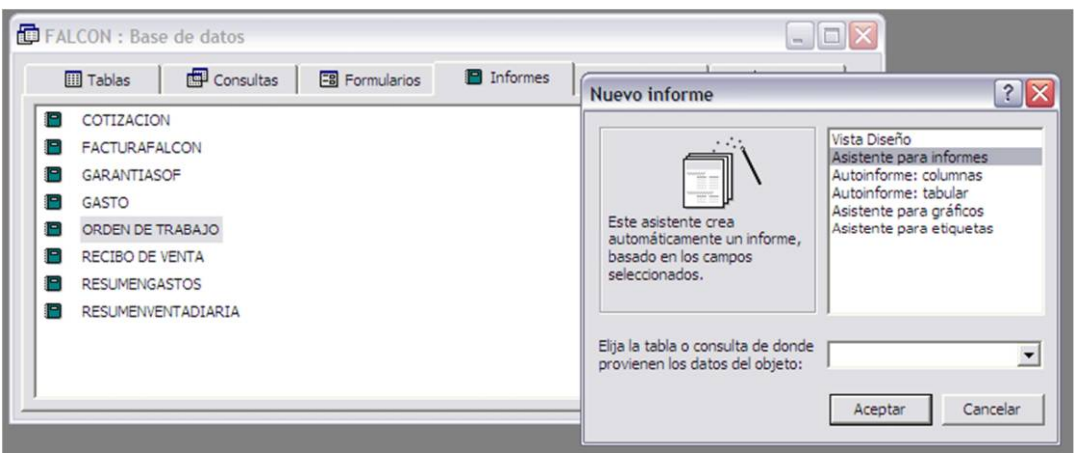

Luego siguiente sale esto



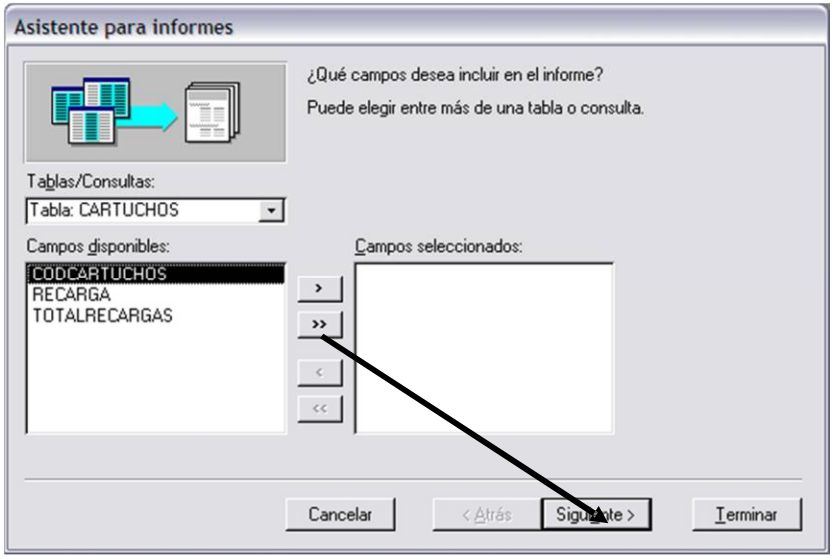
Luego de terminar queda así

Listo para llenar datos

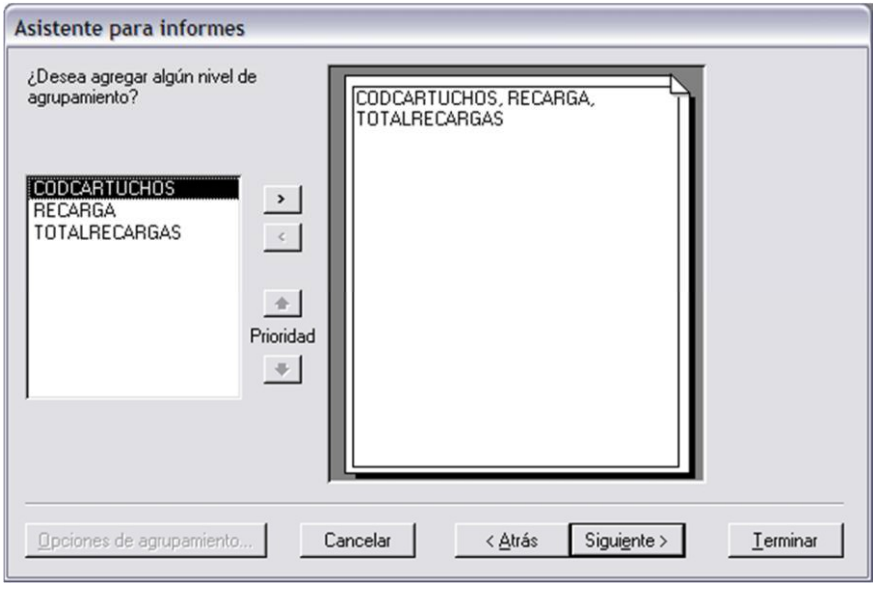
Para elaborar los informes es de la siguiente manera



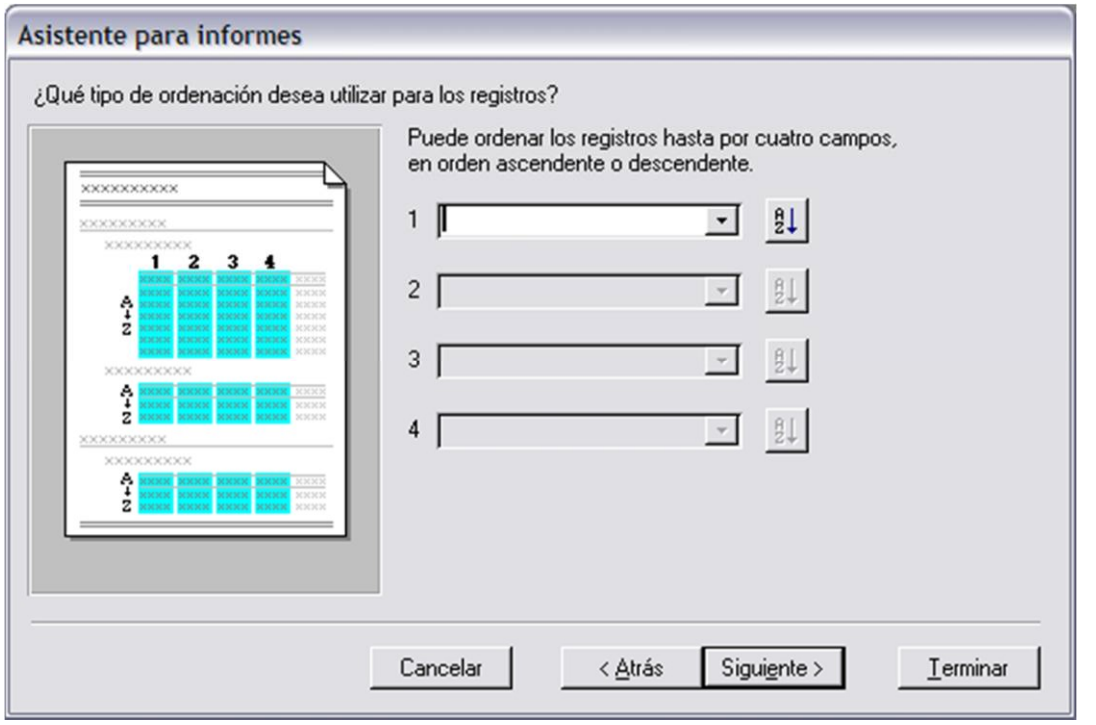
Damos click en la pestaña informes luego asistente para informes y siguiente



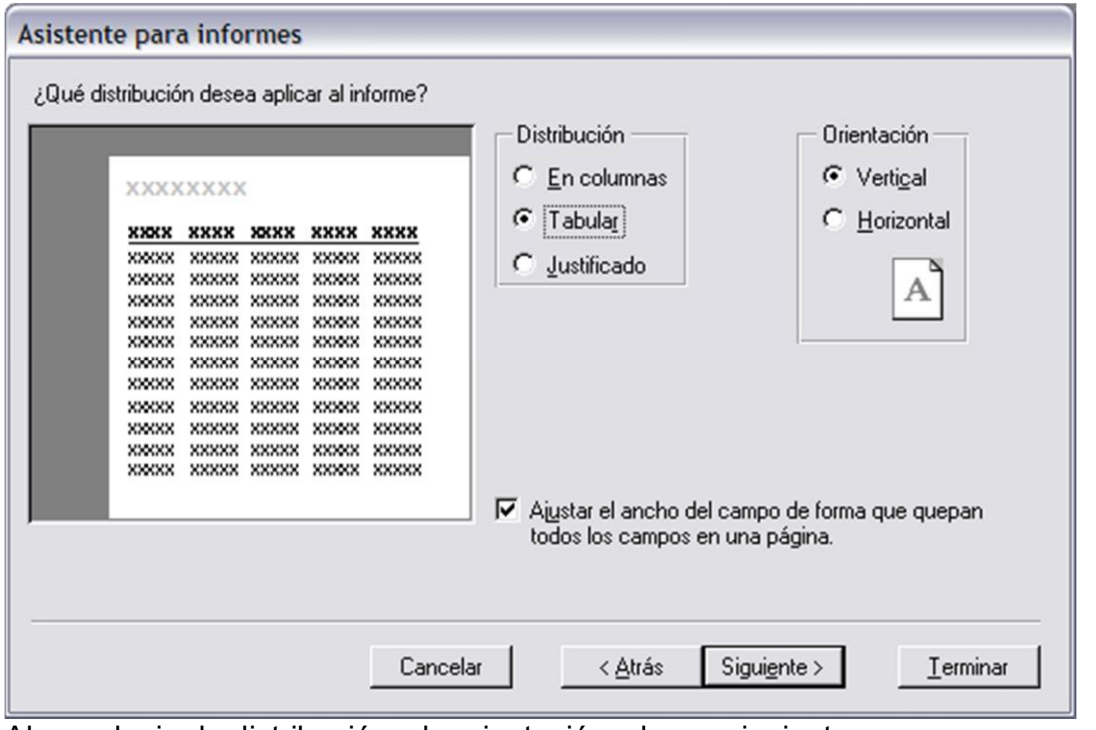

Luego elijo la tabla o entidad a crear el informe agrego los campos y siguiente luego sale otro cuadro simplemente en el doy siguiente



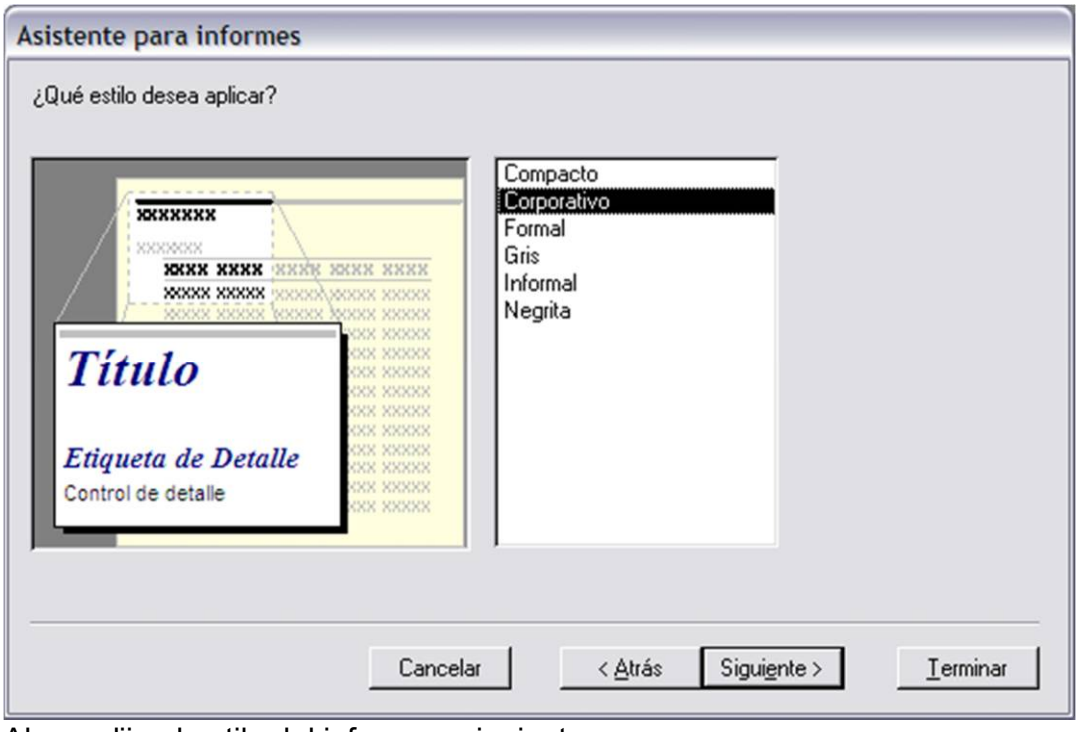

Nuevamente siguiente



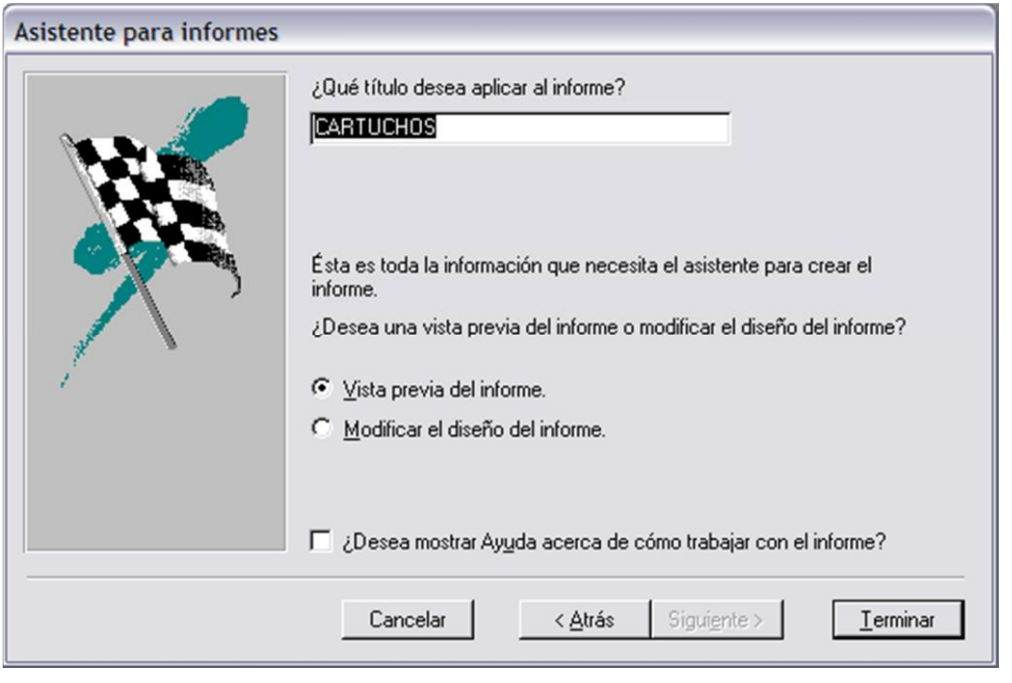

Nuevamente siguiente



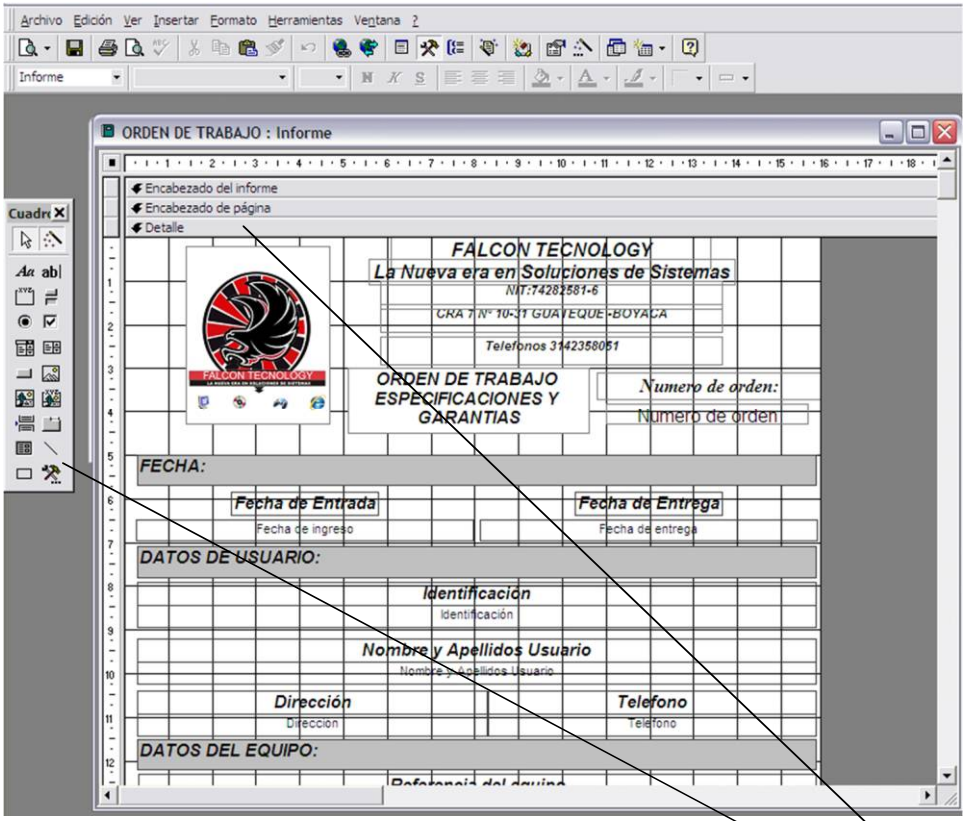

Ahora elegir la distribución y la orientación y luego siguiente



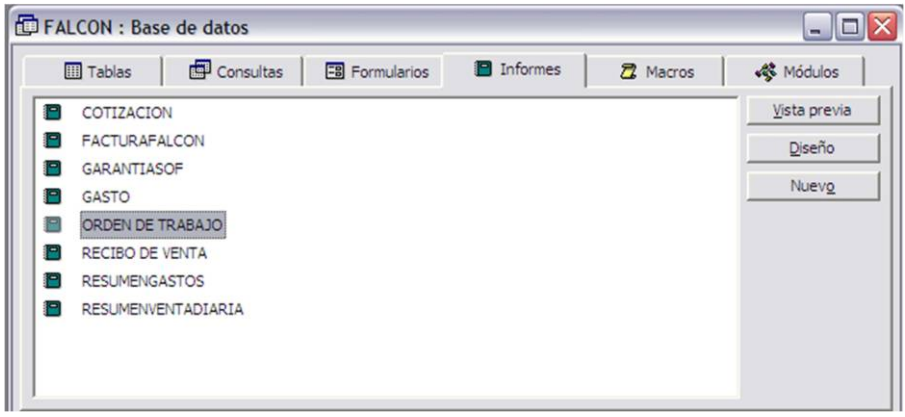

Ahora elijo el estilo del informe y siguiente



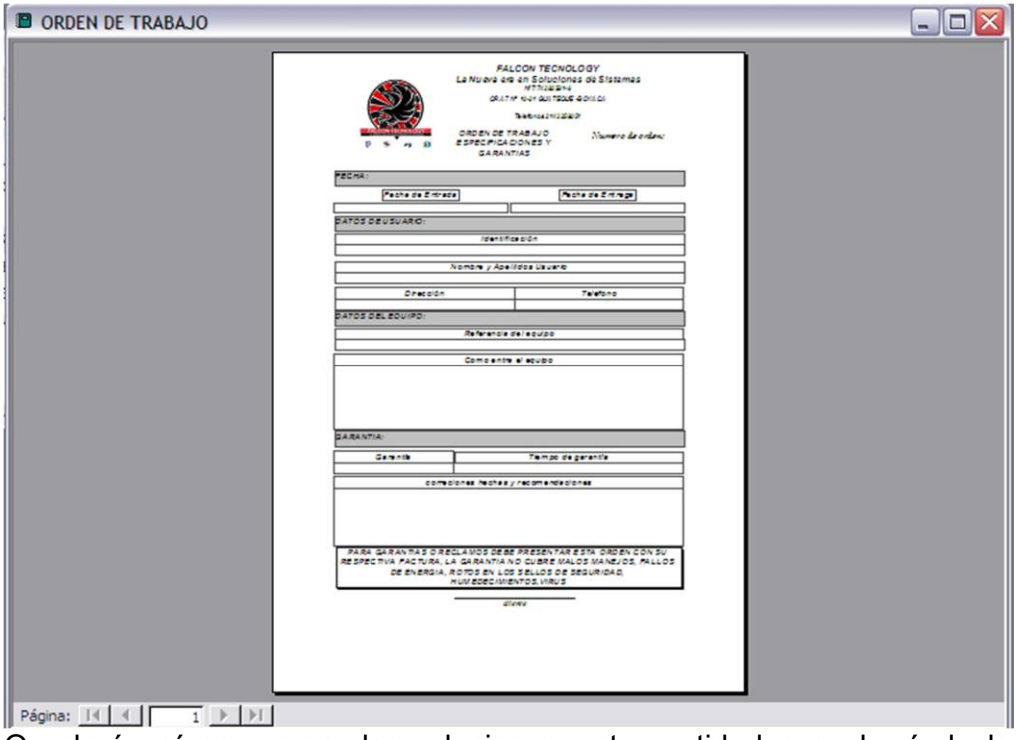
Y ahora le doy terminar queda así para modificarlo si quiero




Hay queda las herramientas para modificarlo en caso de que se necesite para que quede más elegante ya listo quedara así, para que el encabezado se repita en todas las hojas debo mover todo a la opción detalle tal como muestra la gráfica y también debo utilizar el cuadro de controles tal como lo utilizo en visual Basic

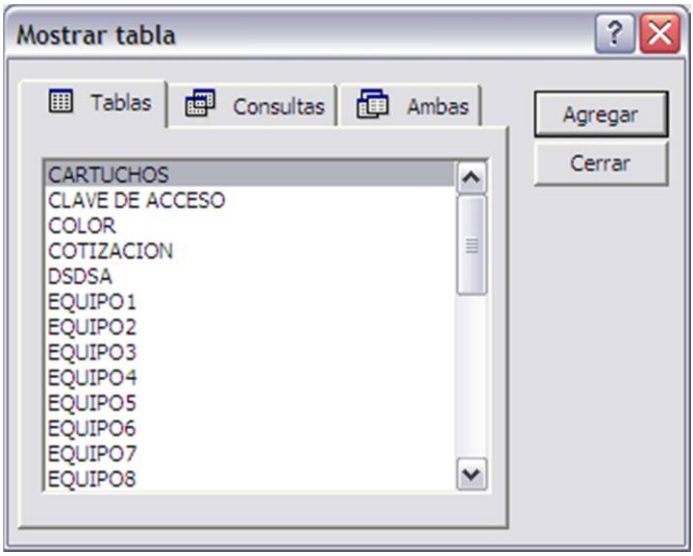



Para verlo click en el botón vista previa

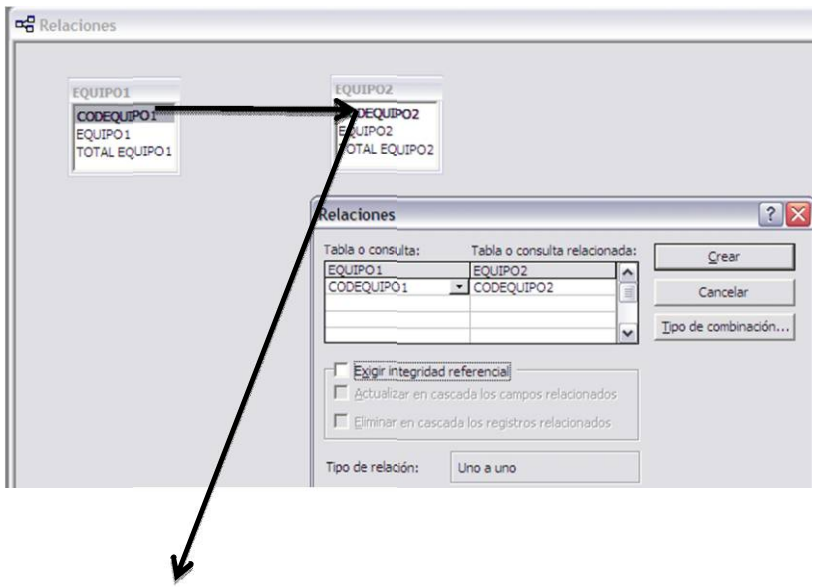



Quedará así para crear las relaciones entre entidades se hará de la siguiente manera

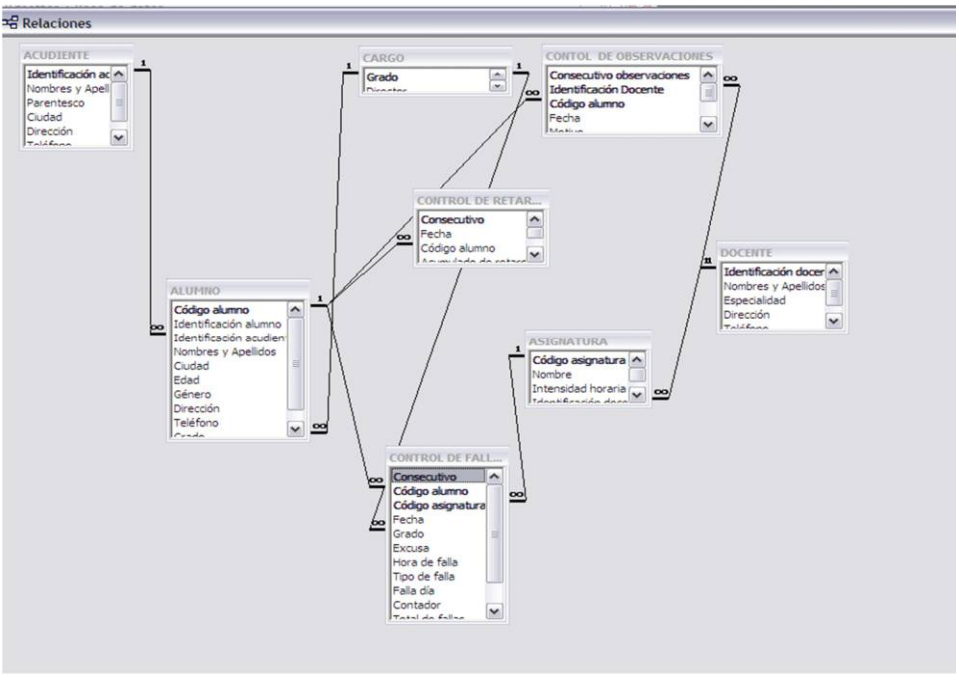
Click aquí




Luego agrego las entidades que necesite relacionar se selecciona y click en el botón agregar luego haremos esto




Arrastro con click sostenido de entidad a entidad que tengan el mismo tipo de datos en la llave principal o clave principal y aparecerá un cuadro, activo la casilla de verificación exigir integridad referencial y luego en el botón crear si sale error están mal creadas las llaves primarias de las entidades para modificar cualquier entidad no debe haber ninguna relación entre ellas cuando ya queden bien se verá algo así




## 26. CONTROL DE HERRAMIENTAS DAO

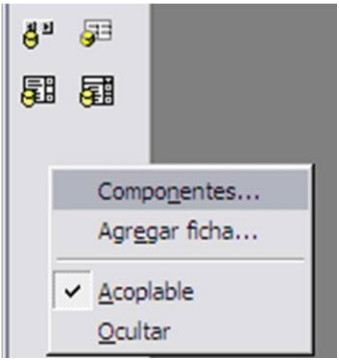
Significa objeto de acceso a datos se enlaza con Access o sqlserver, los controles dao que vamos a utilizar en este proyecto son el famoso control **data** o el control **adodc**

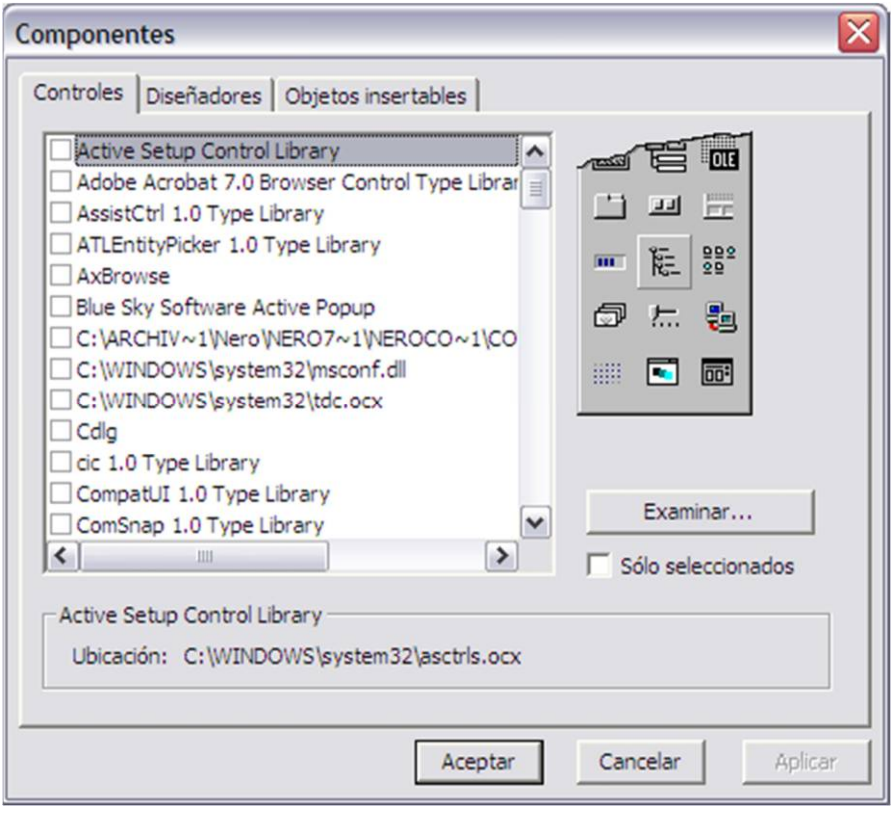
A continuación veremos la imagen de cada control

Para sacar el adodc voy a referencias como se ha explicado anteriormente y cargo esta opción

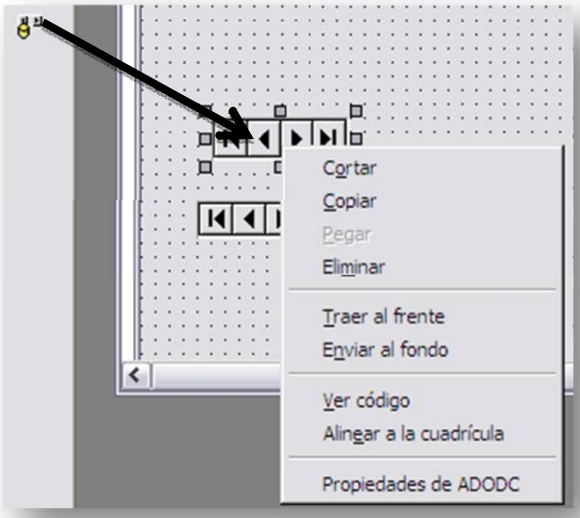
MICROSOFT ADO DATA CONTROL 6.0 (OLEDB)

CONTROL DATACONTROL ADODC

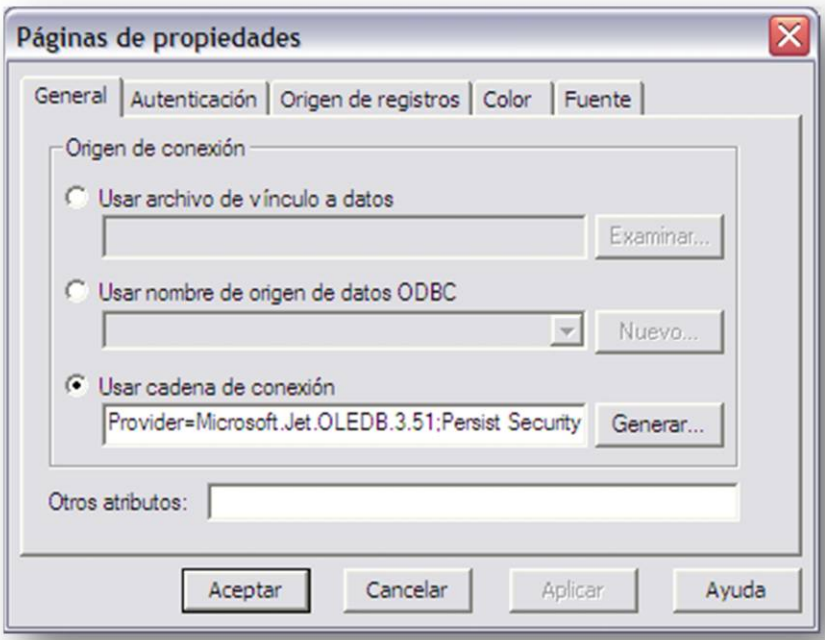


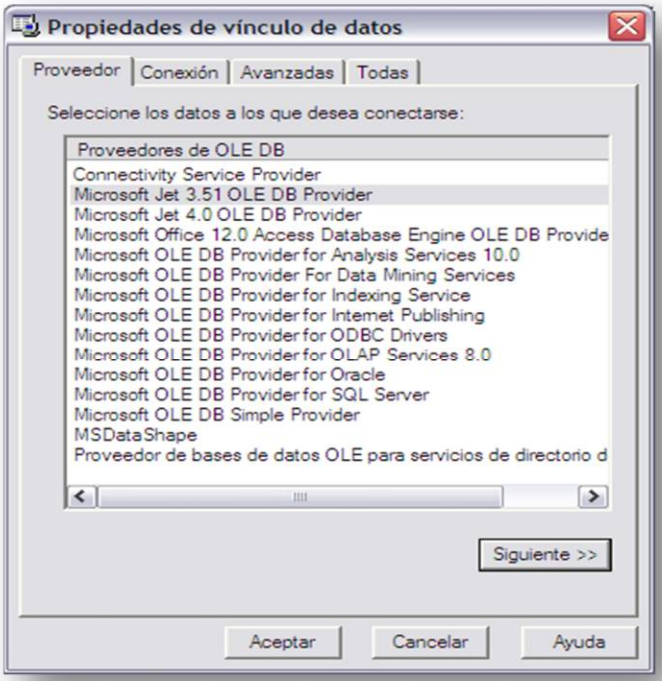

A continuación explicare como conectar un adodc a la base de datos



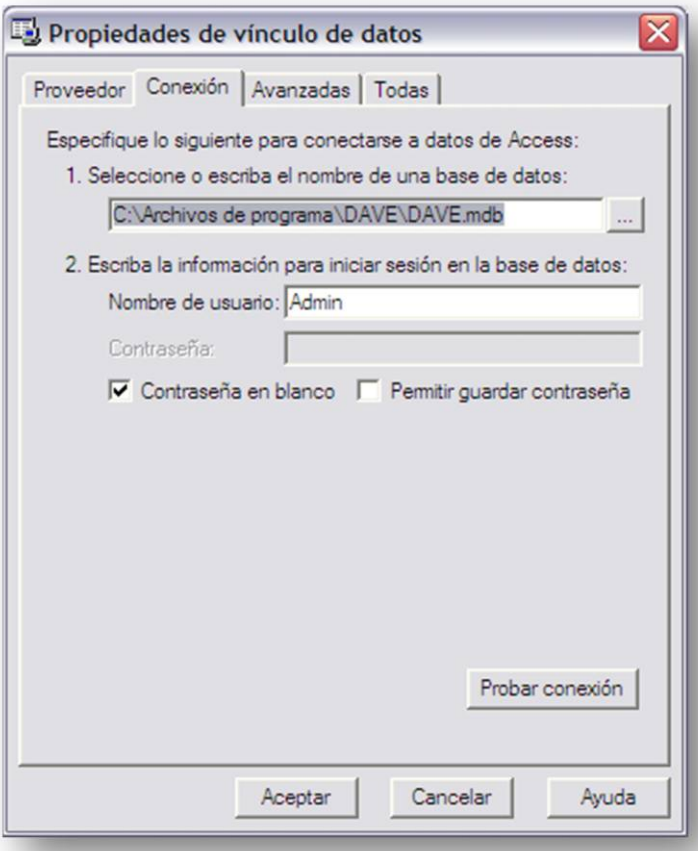

Se dibuja el adodc en cualquier parte del formulario luego damos click derecho y con click normal del mouse propiedades de adodc y aparece lo siguiente

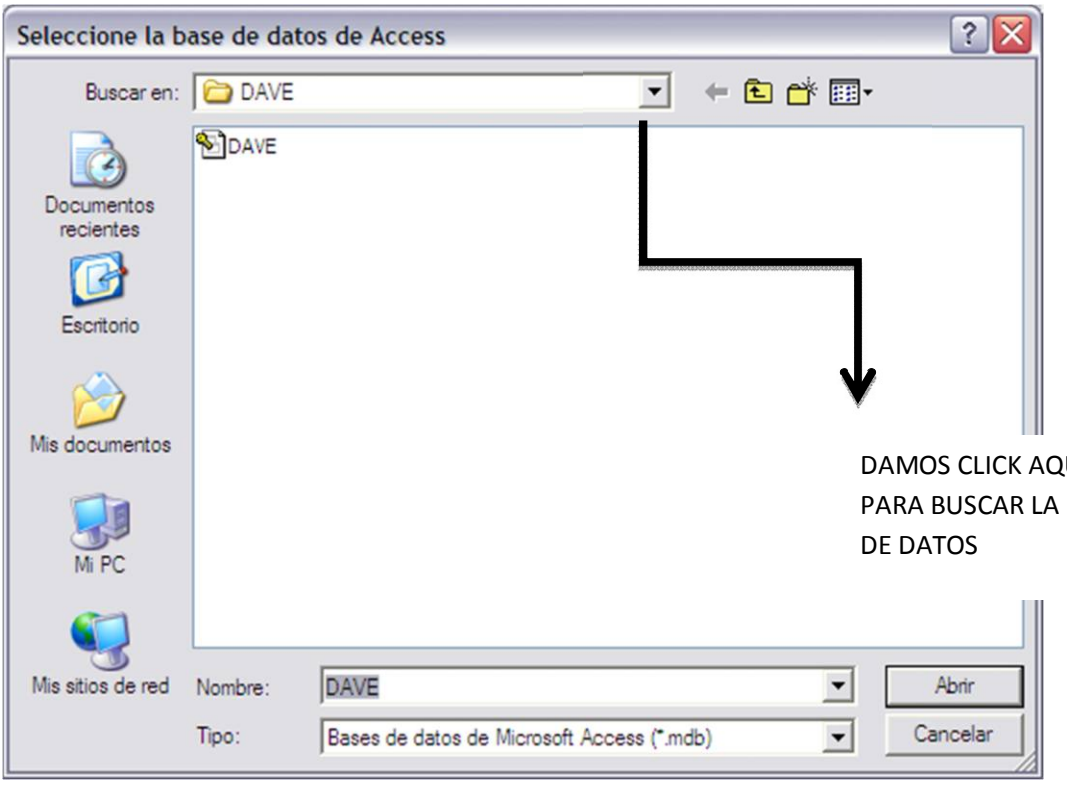



Luego damos click en el botón generar y aparece esto



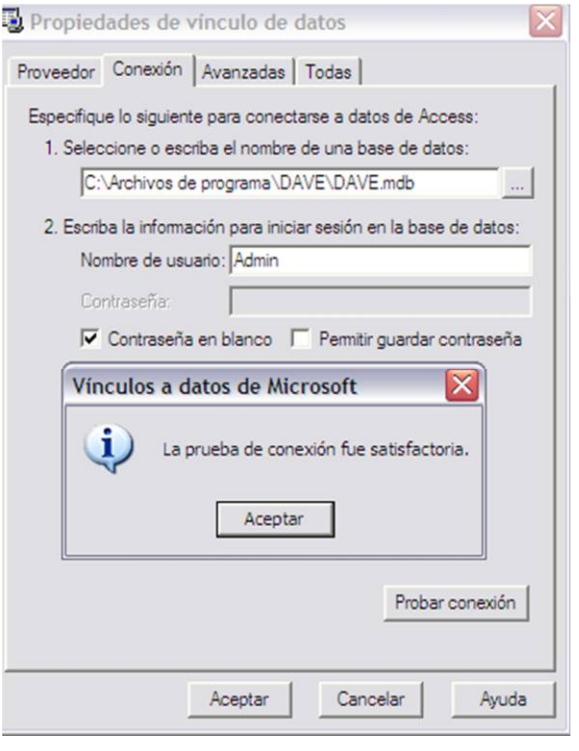

Elegimos la opción Microsoft jet 3.51 ole db provider y luego siguiente

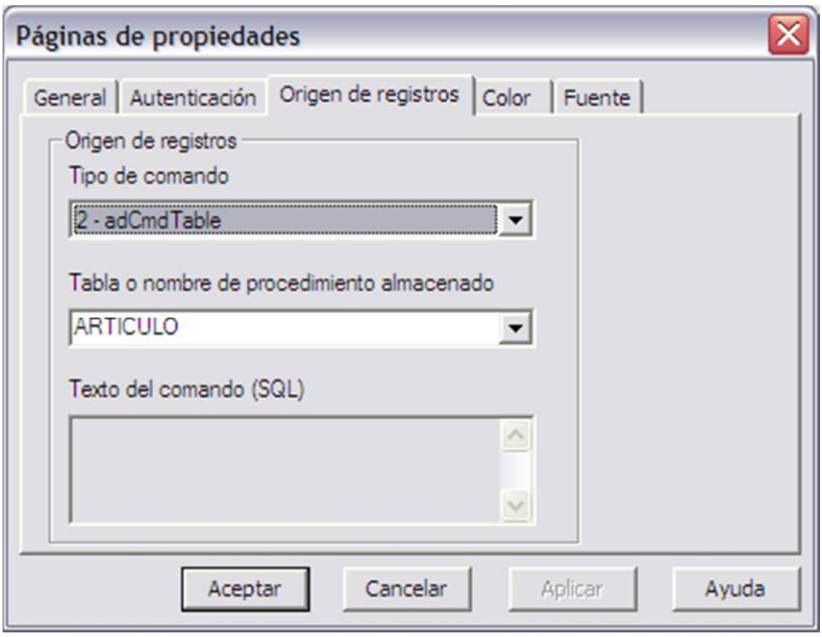





Seleccionamos la base de datos y abrir luego sale esto

Luego le damos click en el botón probar conexión y debe salir este mensaje luego aceptar luego aparece este cuadro



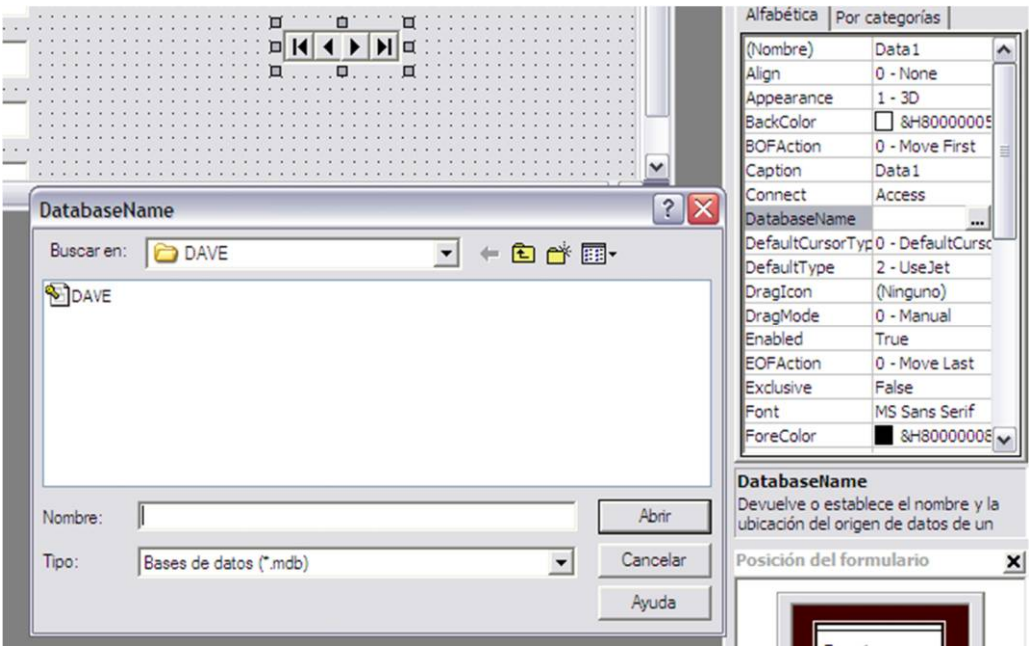



Damos click en la pestaña origen de registros En tipo de comando adcmdtable y debajo el nombre de la entidad o tabla  a conectar y aceptar el adodc ha quedado ya conectado con la base de datos

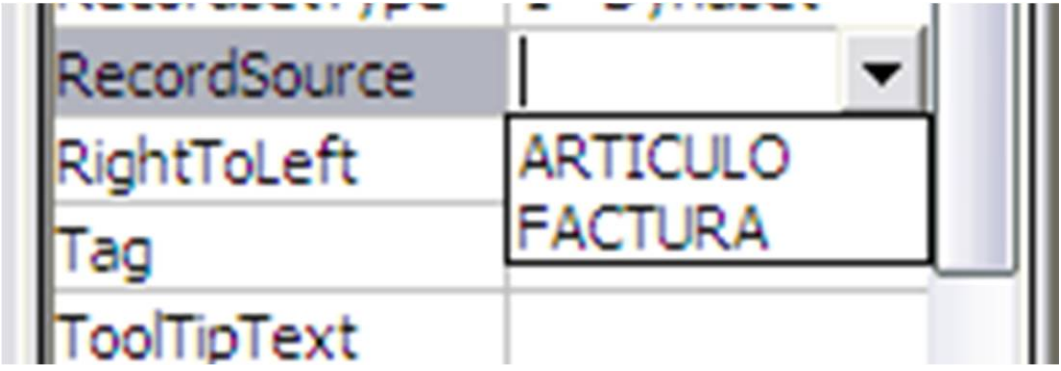
Nota cuando vamos a utilizar datagrid cabe aclarar que ese control solo funciona con adodc no con datas.

A continuación veremos como conectar un data a la base de datos

Dibujo el data luego voy a la ventana de propiedades databasename para conectar con la base de datos luego abrir



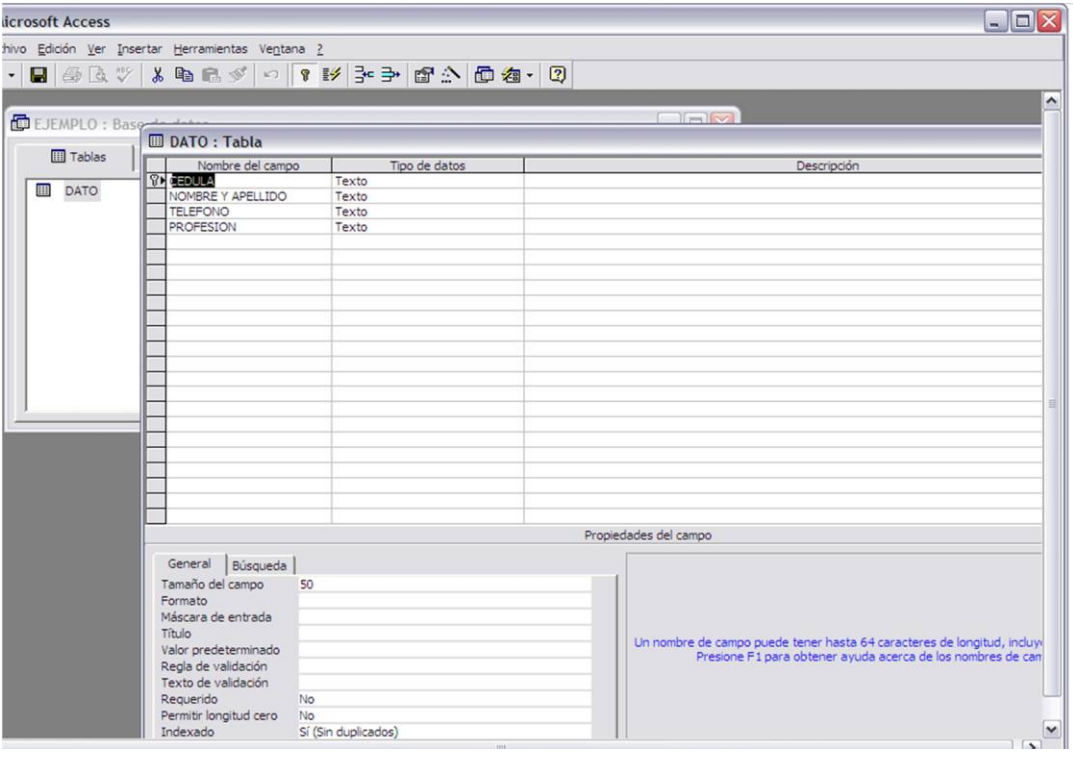
Después hacemos esto



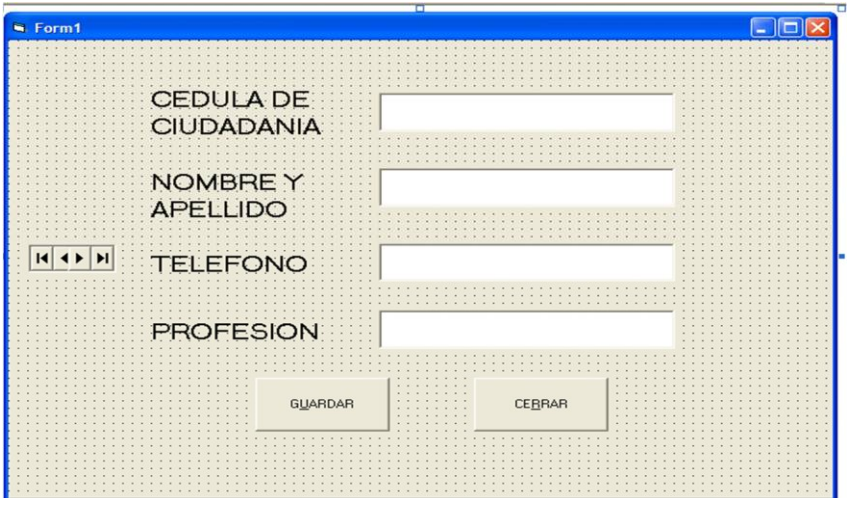

Luego en la propiedad recordsource para conectar con la entidad o tabla dada y listo data ha sido conectada a la base de datos cabe aclarar que después de conectados ya sea un adodc o un data debe quedar en las propiedades visible false, **visible** significa que no quedara viéndose en tiempo de ejecución del software

A continuación ejemplo de botón de guardado en la base de datos

Para este ejemplo haremos una mini base de datos en disco local c: archivos de programa



Necesitamos en este proyecto un data, cajas de texto, labels, botones etc, tal como mostramos en la gráfica debemos conectar el data a la base de datos tal como explique anteriormente en esta guía




La programación de esta aplicación es así

Form | Activate

EVENTO

```
Private Sub Command1_Click()
Data1.Refresh
Data1.Recordset.AddNew
 Data1.Recordset.Fields("CEDULA") = Text1.Text
Data1.Recordset.Fields("NOMBRE Y APELLIDO") = Text2.Text
 Data1.Recordset.Fields("TELEFONO") = Text3.Text
Data1.Recordset.Fields("PROFESION") = Text4.Text
Data1.Recordset.Update
MsgBox "LOS DATOS SE HAN GUARDADO SATISFACTORIAMENTE", vbInformation, "PICHARDO"
Text1.Text = ""
Text2.Text = ""
End Sub

Private Sub Command2_Click()
End
End Sub


Private Sub Form_Activate()
Text1.SetFocus
End Sub
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text2.SetFocus
End If
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text3.SetFocus
End If
End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Text4.SetFocus
End If
End Sub
```

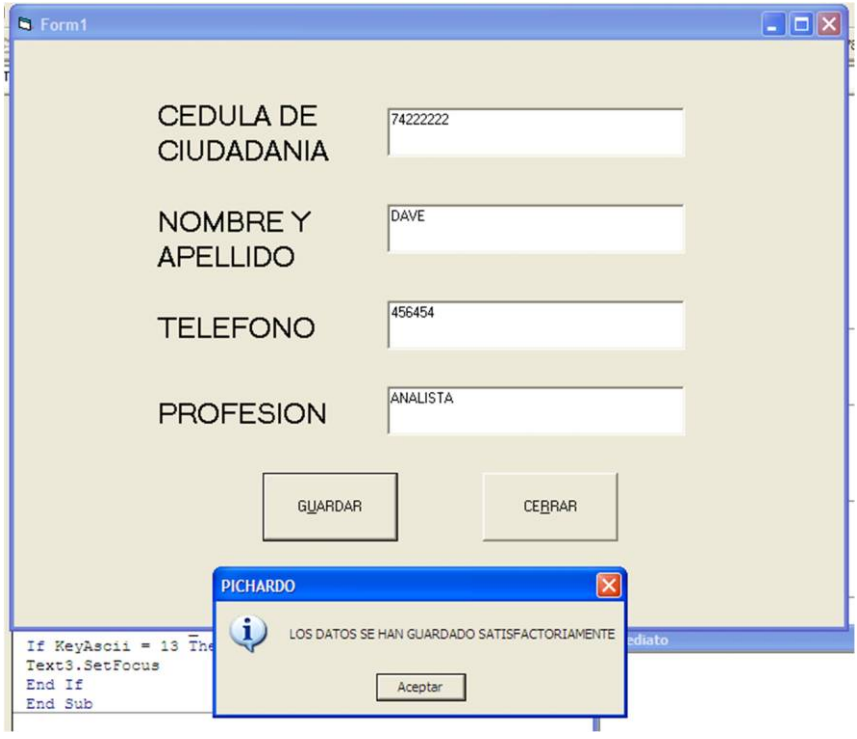
El **13** significa valor que se le asigna a la tecla enter cuando la pulsamos **setfocus** saltar a la otra caja con enter

La propiedad **refresh** que aparece en la programación significa que cada que ingrese un nuevo dato se actualiza o se refresque la base de datos o el data que está conectada a la base de datos para que no se colapse los datos o se encuentren a la hora de guardar **recorset** significa registro **fields** significa nombre del atributo de la entidad por ejemplo
**Data1.recorset.fields (“cedula”)** nombre del atributo que se le asignó a la entidad principal tal como muestro el siguiente grafico cada fields es uno de estos campos

**Addnew** significa adherir nuevo registro cuando decimos**data1.recorset.fields (“cedula”)=text1.text** quiere decir que lo que este en esa caja lo guarde en ese campo o atributo de esa entidad
**Msgbox "los datos se han guardado satisfactoriamente", vbinformation, "pichardo"** esta instrucción muestra un mensaje al usuario después de guardar con el nombre que se le dio al software la instrucción **update** significa guardar la instrucción **text1.text=””** significa que después de guardar vacie las cajas para llenar nuevos datos en la propiedad **activate del formulario text1.setfocus** significa que cuando ejecutemos el programa el cursor del mouse aparezca en esa caja para programar en cada control independientes se debe dar doble click y elegir el evento a utilizar

```
Private sub text1_keypress(keyascii as integer)
If keyascii = 13 then
Text2.setfocus
End if
End sub
```

Esta instrucción significa que cada que de enter saltara a la caja correspondiente para llenar datos en ejecución **end sub** significa fin de la instrucción quedara así



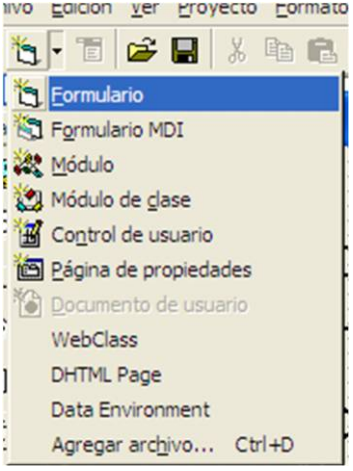

Al darle aceptar tuvo que haber guardado ya en la base de datos tal como muestra en la gráfica

Para ver si guardo o no, abrimos la base de datos y luego elegimos tabla, seleccionamos la entidad y le damos en el botón abrir.

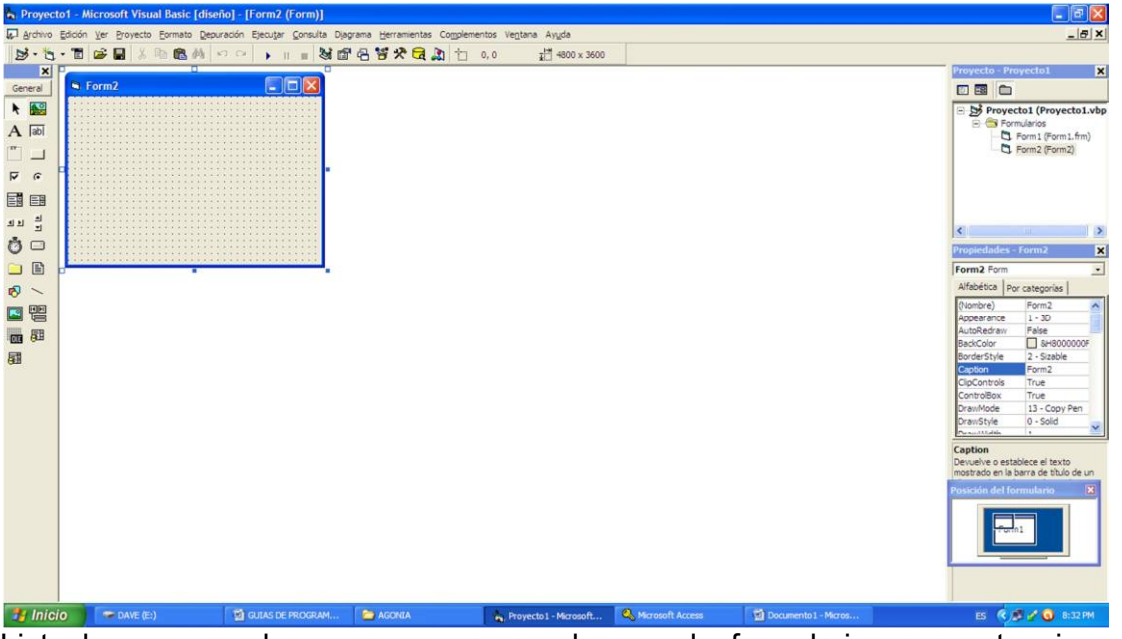
El ejercicio de la guía es hacer una aplicación parecida a este pero con artículos de la canasta familiar para ello debo crear una base de datos para llenar los artículos

Ejemplo siguiente es cargar datos existentes de la base de datos a las cajas de texto utilizando el control **dbcombo** para ello sacare componentes como se explica anteriormente y elegimos esta opción **Microsoft data bound list controls 6.0**

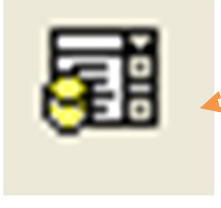
Para este ejemplo utilizamos parte del ejercicio anterior pero cargándole otro formulario al proyecto de esta forma



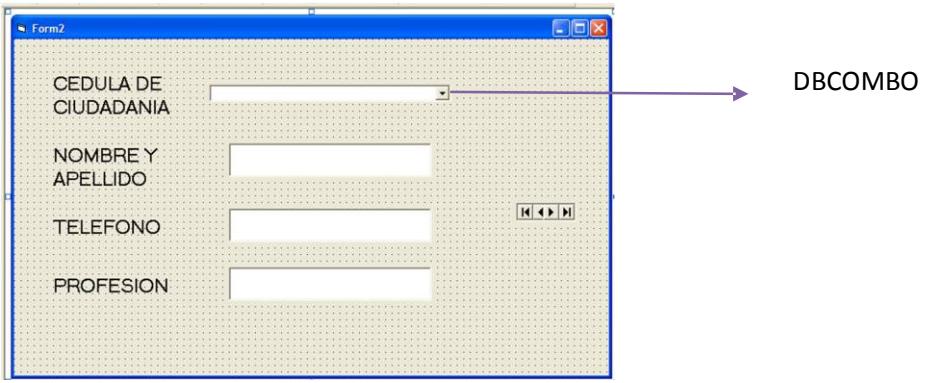

Quedará así el área




Listo luego procedemos a programar el segundo formulario para este ejemplo utilizare data que deben conectar a la base de datos tal como explique anteriormente dbcombo,cajas de texto y labels o etiquetas



Para probar la ejecución del nuevo formulario debemos cambiar el orden de ejecución en propiedades tal como explique anteriormente en esta guía




Después de haber conectado el data a la base de datos tal como aprendimos anteriormente vamos a conectar el combo al data y a la entidad respectiva utilizando las siguientes propiedades

**Rowsourse** y elijo el data Luego **data source** y elijo el data Luego **listfields** y elijo el campo que se va a cargar en este caso las cedula guardadas luego procedo a programar en el combo dándole doble click
Todas las cajas de texto irán **enableb=false** para que no se pueda modificar el contenido de las cajas de texto pero modificando el **font** de las cajas hará que la letra quede más grande para borrar el nombre que aparece en el **dbcombo** acudimos a la propiedad **text** para borrar ese contenido la programación de este ejemplo quedara así.

DBCombo1

```
Private Sub DBCombo1_Click(Area As Integer)
On Error Resume Next
If Err.Number = 3021 Then
End If
Data1.Refresh
Data1.Recordset.MoveFirst
Do While Not Data1.Recordset.EOF
If Data1.Recordset.Fields("CEDULA") = DBCombo1.Text Then
Exit Do
Else
Data1.Recordset.MoveNext
End If
Loop
Text2.Text = Data1.Recordset("NOMBRE Y APELLIDO")
Text3.Text = Data1.Recordset("TELEFONO")
Text4.Text = Data1.Recordset("PROFESION")
End Sub
```

Como algunas veces al programar visual nos arroja un error este se puede obviar de la siguiente manera colocando esta instrucción y según el número del error donde corresponda

**On error resume next**
**If err.number = 3021 then**
**End if**

Esta instrucción significa que el data se actualice o se refresque

**Data1.refresh**

Esta instrucción significa que la base de datos se mueva al primer registro guardado en la entidad

**Data1.recordset.movefirst**

Para el último registro es

**Data1.recordset.movelast**

Esta instrucción significa que mientras que el data con su registro no se el ultimo valor me muestre la cedula en el dbcombo y los valores guardados posteriormente en la base de datos

**Eof=ultimo valor o registro**
**Bof=primer valor o registro**

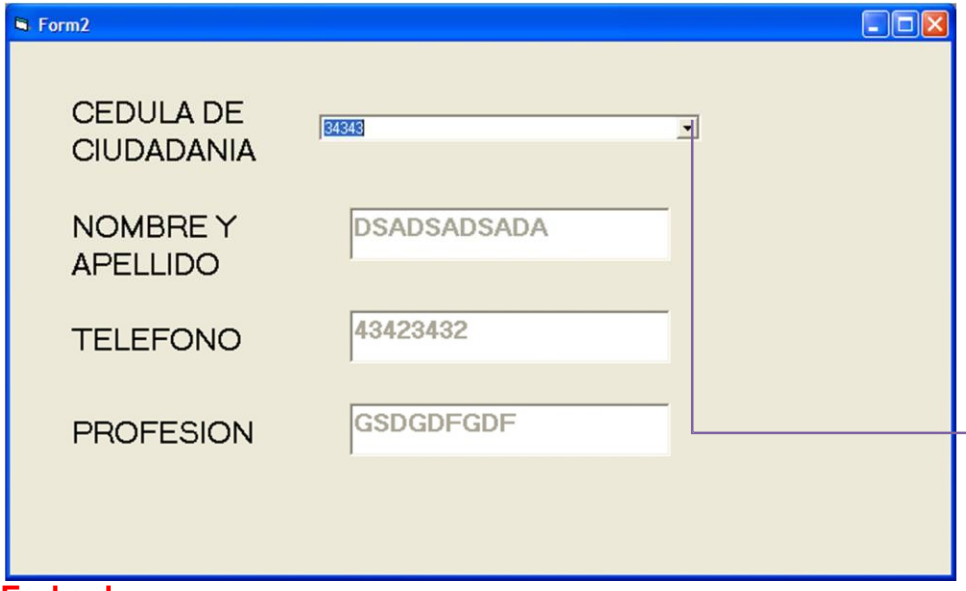
**Do while not data1.recordset.eof**
**If data1.recordset.fields ("cedula") = dbcombo1.text then**
**Exit do**

Esta instrucción significa que si la cedula no corresponde al dato que se mueva al siguiente registro para buscar los valores

**Else**
**Data1.recordset.movenext**
**End if**

Esta instrucción significa que cada vez que cargue el valor correcto muestre los valores en las cajas las veces que sea
Cada **recordset** es cada campito o atributo de la entidad creada va mostrar lo que este hay guardado

**Loop**
**Text2.text = data1.recordset("nombre y apellido")**
**Text3.text = data1.recordset("telefono")**
**Text4.text = data1.recordset("profesion")**

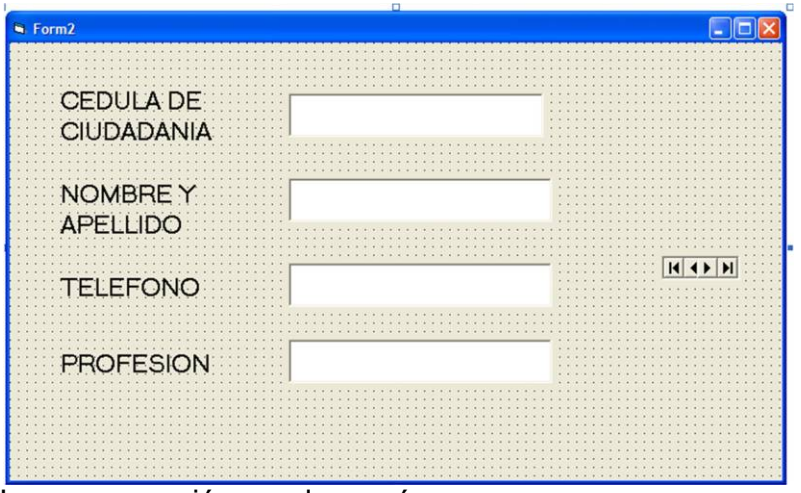



**End sub**

En tiempo de ejecución quedara así

Para que estos valores carguen en este dbcombo doy click hay y arrastrando el mouse sin soltar elijo el valor o si no quedara seleccionado

El ejercicio de este ejemplo es hacer uno parecido pero no utilizar dbcombo si no en una caja de texto digitar el valor y que me muestre los datos utilizando el evento **keypress** de la caja de texto sin embargo aquí debajito les dejo la programación de como se hace si fuera una caja de texto



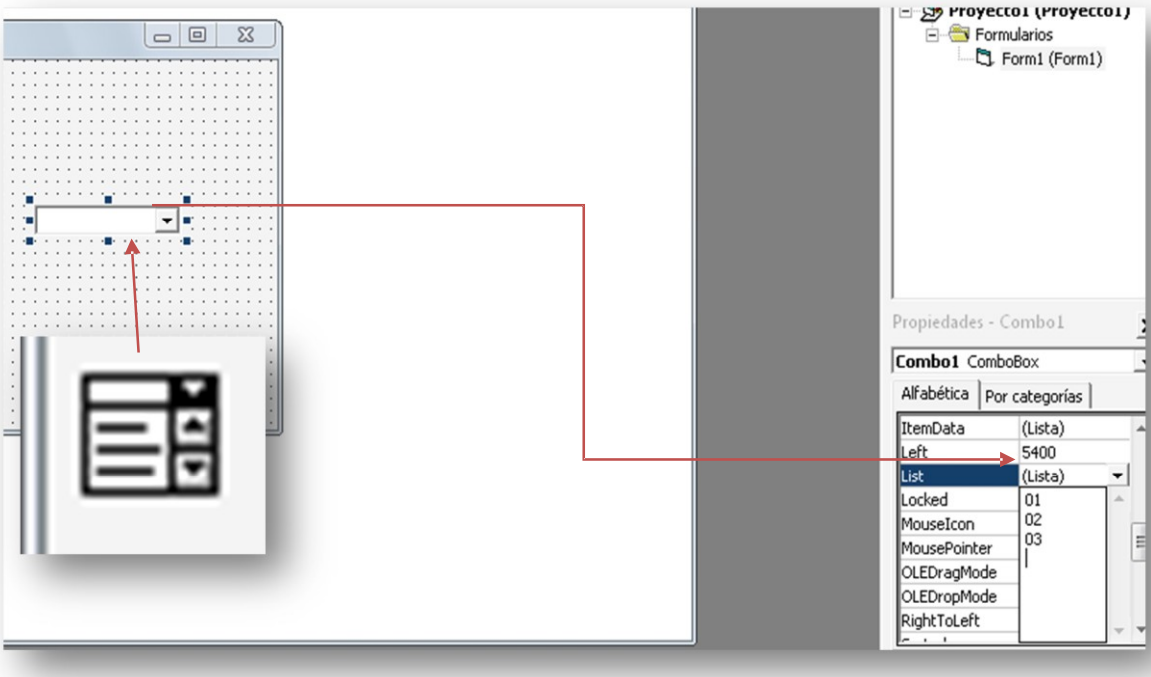

La programación quedara así

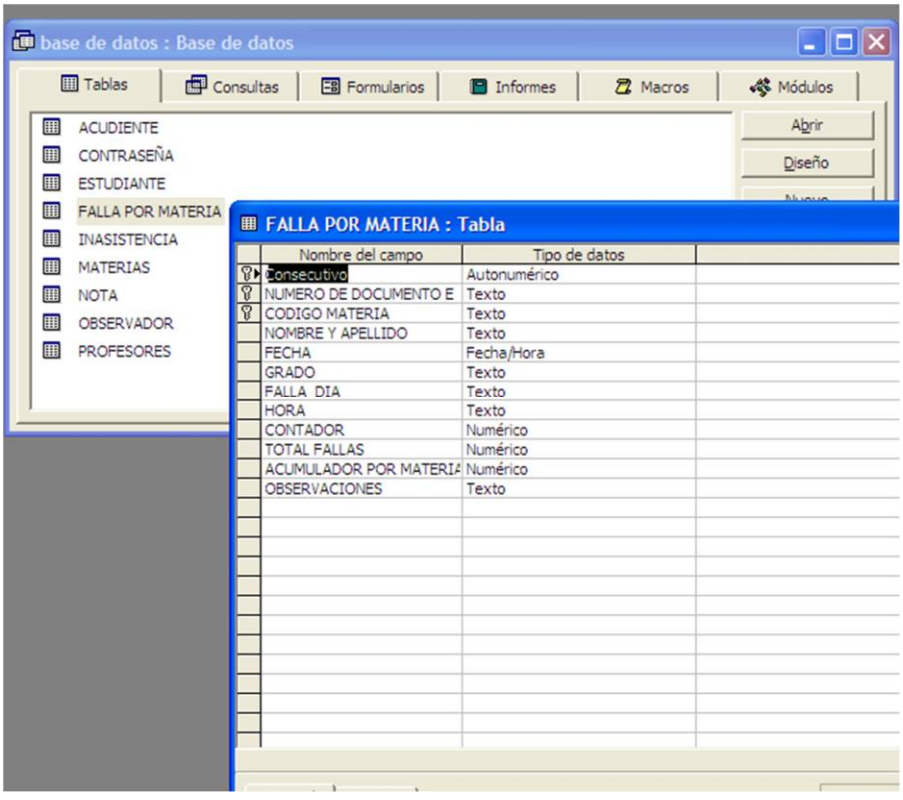
Text1 | KeyPress

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
On Error Resume Next
If Err.Number = 3021 Then
End If
Data1.Refresh
Data1.Recordset.MoveFirst
Do While Not Data1.Recordset.EOF
If Data1.Recordset.Fields("CEDULA") = Text1.Text Then
Exit Do
Else
Data1.Recordset.MoveNext
End If
Loop
Text2.Text = Data1.Recordset("NOMBRE Y APELLIDO")
Text3.Text = Data1.Recordset("TELEFONO")
Text4.Text = Data1.Recordset("PROFESION")
End If
End Sub
```

# 27. BOTON DE BORRADO DE REGISTRO UNO A UNO

En este ejemplo también para a utilizar combobox para cargarle los datos al combo utilizamos la propiedad **list** y para dar enter o escribir más valores tocar ir dando control enter si no se cierra y no deja seguir escribiendo valores a la vez. Ejemplo de acumuladores de acuerdo a un serial para ello creamos una base de datos como vamos a mostrar a continuación





Elaborar un formulario con estas especificaciones comnand button dbcombox combo box y cajas de texto

La programación de todo esto es así

La **variable bandera** cuando está en true activa toda la programación que este allí

**Variables**

```
Dim acumulador, total, contador, r, bandera, c as integer
Dim fecha  as string
```

---

```
Private sub cmborrar_click() 4
Fecha = cbdía.text + "/" + cbmes.text + "/" + cbaño.text esto significa concatenar datos para que guarde en acces así   01/02/2011
If dbcombo1.text = "" or dbcombo2.text = "" or cbdía.text = "" or cbmes.text = "" or cbaño.text = "" then
Msgbox "digite el código del alumno y de la asignatura que desea borrar y la fecha correspondiente a esa falla"
Else
Dtfalla.refresh
Dtfalla.recordset.movefirst
Do while not dtfalla.recordset.eof
If dtfalla.recordset.fields("numero de documento e") = dbcombo1.text and dtfalla.recordset.fields("fecha") = fecha and dtfalla.recordset.fields("codigo materia") = dbcombo1.text then
Exit do
Else
Dtfalla.recordset.movenext
End if
Loop
Dtfalla.refresh
Dtfalla.recordset.delete delete significa borrar datos
End if
Dbcombo1.text = ""
Dbcombo2.text = ""
Cbdía.text = ""
Cbmes.text = ""
Cbaño.text = ""
Txnom.text = ""
Txno.text = ""
Text10.text = ""
Txhfa.text = ""
Cbfallasdía.text = ""
Txacu.text = ""
Txtfa.text = ""
Txgra.text = ""
End sub
```

---

```
Private sub cmcerrar_click() 5
Beep
Unload me
End sub
```

---

```
Private sub cmguardar_click() 3
```

```
Control = "falla"
Bandera = true
Fecha = cbdía.text + "/" + cbmes.text + "/" + cbaño.text
If cbdía.text = "" or cbmes.text = "" or cbaño.text = "" or dbcombo1.text = "" or
dbcombo2.text = "" or txhfa.text = "" or cbfallasdía.text = "" then
Msgbox "no deje campos vacíos", vbcritical, "error en la base de datos"
Else
Dtfalla.refresh
If dtfalla.recordset.eof = true then
Txacu = cbfallasdía.text
Txtfa = cbfallasdía.text
C = cbfallasdía.text
Dtfalla.refresh
Dtfalla.recordset.addnew
Dtfalla.recordset.fields("numero de documento e") = dbcombo1.text
Dtfalla.recordset.fields("nombre y apellido") = txnom.text
Dtfalla.recordset.fields("codigo materia") = dbcombo2.text
Dtfalla.recordset.fields("fecha") = fecha
Dtfalla.recordset.fields("grado") = txgra.text
Dtfalla.recordset.fields("hora") = txhfa.text
Dtfalla.recordset.fields("falla  dia") = cbfallasdía.text
Dtfalla.recordset.fields("total fallas") = cbfallasdía.text
Dtfalla.recordset.fields("acumulador por materia") = cbfallasdía.text
Dtfalla.recordset.fields("contador") = c
Dtfalla.recordset.fields("observaciones") = text10.text
Dtfalla.recordset.update
Else
Dtfalla.refresh
Dtfalla.recordset.movelast esto significa que se mueva al ultimo valor
Do while not dtfalla.recordset.eof
If dtfalla.recordset.fields("numero de documento e") = dbcombo1.text and
dtfalla.recordset.fields("codigo materia") = dbcombo2.text then
Exit do
Else
'Manejo de error base de datos
On error resume next
If err.number = 3021 then
Dtfalla.recordset.addnew
Dtfalla.recordset.fields("numero de documento e") = dbcombo1.text
Dtfalla.recordset.fields("nombre y apellido") = txnom.text
Dtfalla.recordset.fields("codigo materia") = dbcombo2.text
Dtfalla.recordset.fields("fecha") = fecha
Dtfalla.recordset.fields("grado") = txgra.text
Dtfalla.recordset.fields("hora") = txhfa.text
Dtfalla.recordset.fields("falla dia") = cbfallasdía.text
Dtfalla.recordset.fields("total fallas") = cbfallasdía.text
Dtfalla.recordset.fields("acumulador por materia") = cbfallasdía.text
```

```
Dtfalla.recordset.fields("contador") = c
Dtfalla.recordset.fields("observaciones") = text10.text
Dtfalla.recordset.updatedtfalla name o nombre que se le dio al data para
programar
Exit sub
Endif
' fin manejo de error
Dtfalla.recordset.moveprevious esto significa que se mueva al anterior
Endif
Loop
Acumulador = dtfalla.recordset.fields("acumulador por materia")
Txacu.text = acumulador + val(cbfallasdía.text)
Dtfalla.refresh
Dtfalla.recordset.movelast
Do while not dtfalla.recordset.eof
If dtfalla.recordset.fields("numero de documento e") = dbcombo1.text then
Exit do
Else
Dtfalla.recordset.moveprevious
End if
Loop
Contador = dtfalla.recordset.fields("contador")
Bandera = true
If contador>= 3 then
R = (contador - 3) + val(cbfallasdía.text)
End if
Dtfalla.refresh
Dtfalla.recordset.movelast
Do while not dtfalla.recordset.eof
If dtfalla.recordset.fields("numero de documento e") = dbcombo1.text then
Exit do
Else
Dtfalla.recordset.moveprevious
End if
Loop
Total = dtfalla.recordset.fields("total fallas")
Txtfa.text = total + val(cbfallasdía.text)
Dtfalla.refresh
Dtfalla.recordset.addnew
Dtfalla.recordset.fields("numero de documento e") = dbcombo1.text
Dtfalla.recordset.fields("nombre y apellido") = txnom.text
Dtfalla.recordset.fields("codigo materia") = dbcombo2.text
Dtfalla.recordset.fields("fecha") = fecha
Dtfalla.recordset.fields("grado") = txgra.text
Dtfalla.recordset.fields("hora") = txhfa.text
Dtfalla.recordset.fields("falla  dia") = cbfallasdía.text
Dtfalla.recordset.fields("total fallas") = total + val(cbfallasdía.text)
```

```
Dtfalla.recordset.fields("acumulador por materia") = acumulador +
val(cbfallasdía.text)
Dtfalla.recordset.fields("contador") = c
Dtfalla.recordset.fields("observaciones") = text10.text
Dtfalla.recordset.update
End if
End if
End sub
```

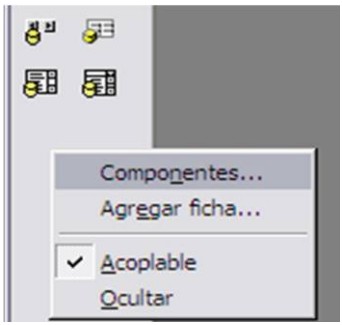
---

```
Private sub dbcombo1_click(area as integer) 1
On error resume next
If err.number = 3021 then
End if
Dtalu.refresh
Dtalu.recordset.movefirst
Do while not dtalu.recordset.eof
If dtalu.recordset.fields("numero de documento e") = dbcombo1.text then
Exit do
Else
Dtalu.recordset.movenext
End if
Loop
Txnom.text = dtalu.recordset("nombre y apellido")
Txgra.text = dtalu.recordset("grado")

End sub

Private sub dbcombo2_click(area as integer) 2
On error resume next
If err.number = 3021 then
End if
Dtasi.refresh
Dtasi.recordset.movefirst
Do while not dtasi.recordset.eof
If dtasi.recordset.fields("codigo materia") = dbcombo2.text then
Exit do
Else
Dtasi.recordset.movenext
End if
Loop
Txno.text = dtasi.recordset("nombre de materia")
Cbdía.setfocus
End sub
```

---

Adodc y data grid para hacer este ejemplo crearemos una mini base de datos de artículos y ventas de productos

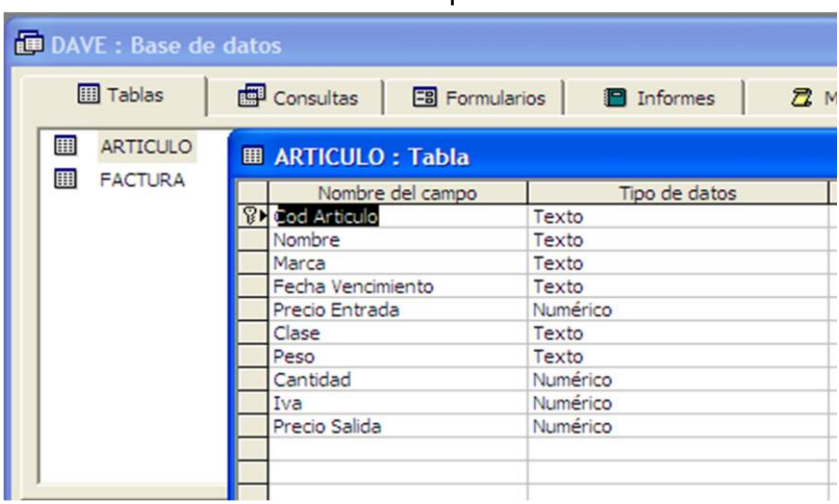
Sacar en componentes



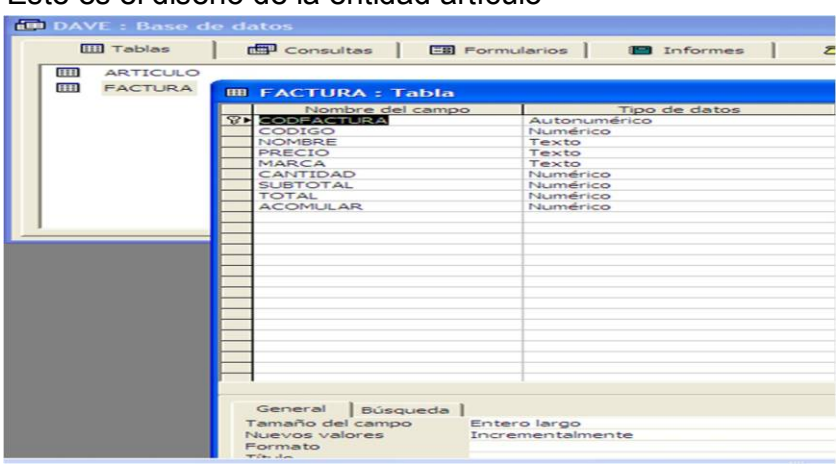

**MICROSOFT DATA GRID COTROL 6.0 (OLEDB)**

**MICROSOFT ADO DATA CONTROL 6.0 (OLEDB)**

A continuación mostrare como queda la mini base de datos



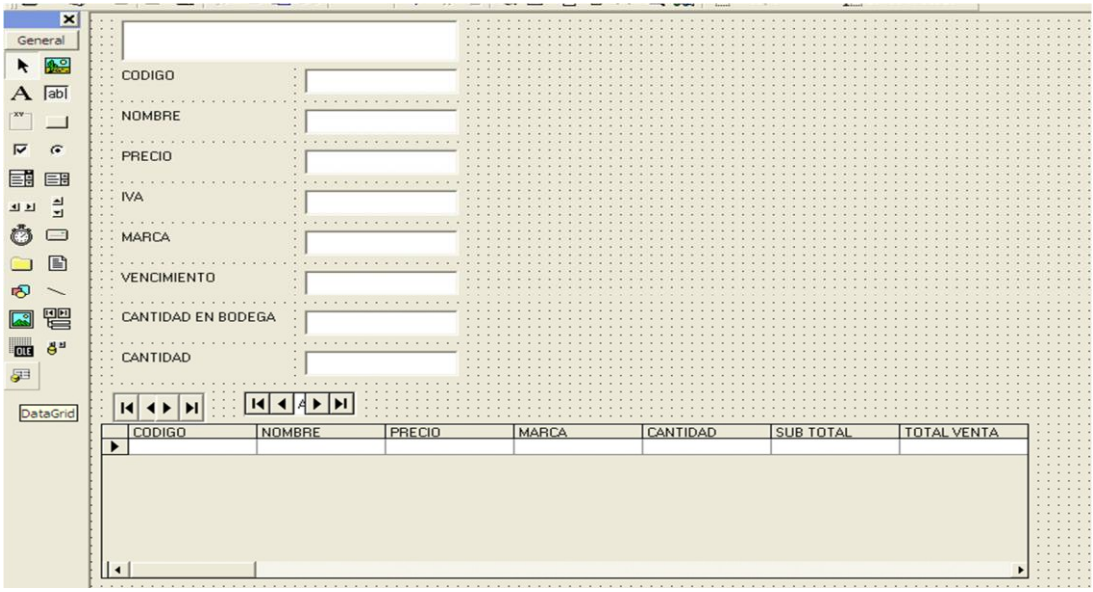

Este es el diseño de la entidad artículo



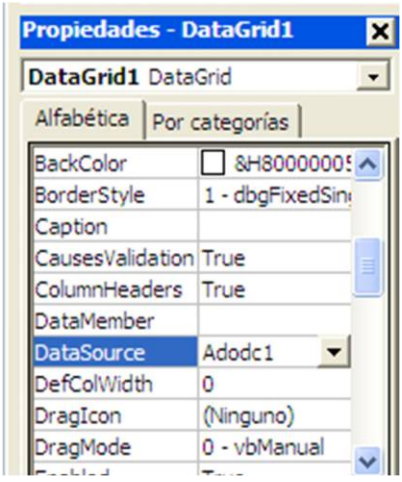
Este es el diseño de la entidad factura

Lo primero que vamos hacer es conectar el adodc a la base de datos a la tabla artículo tal como explique anteriormente

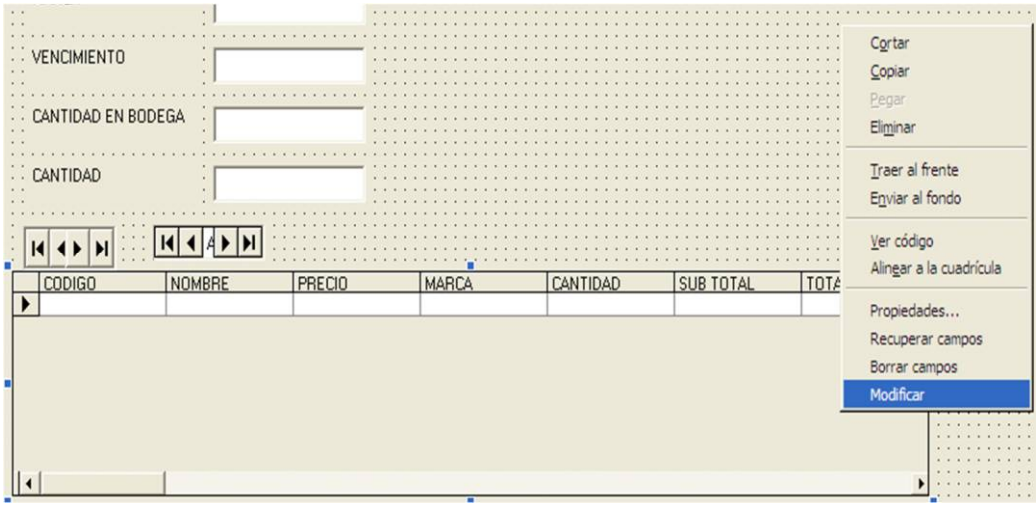
Después de conectado el adodc vamos a conectar el **datagrid** al adodc para que a la hora de guardar datos se visualizan en la grilla



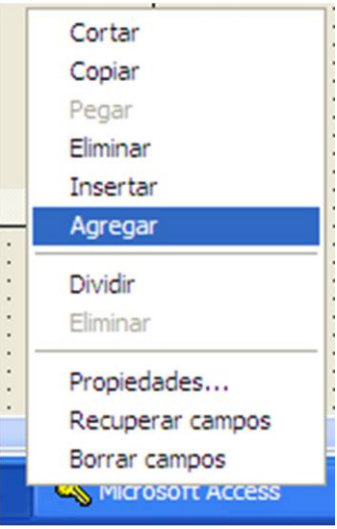

Seleccionamos la grilla utilizamos la propiedad data source para conectar el data grid al adodc



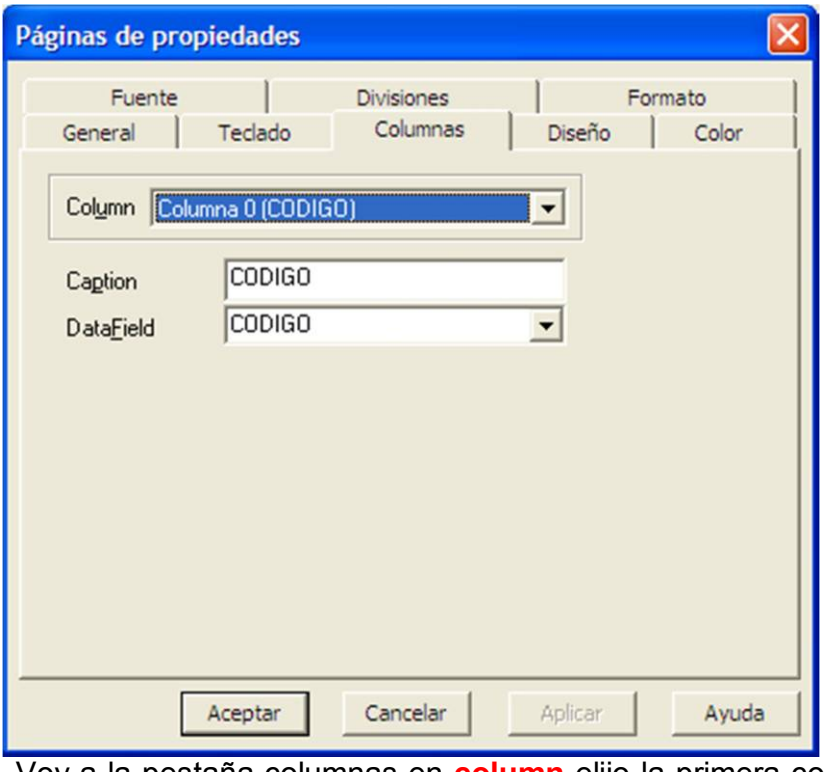

Ahora veremos como colocarle los caption o los nombres de cada campo a la grilla



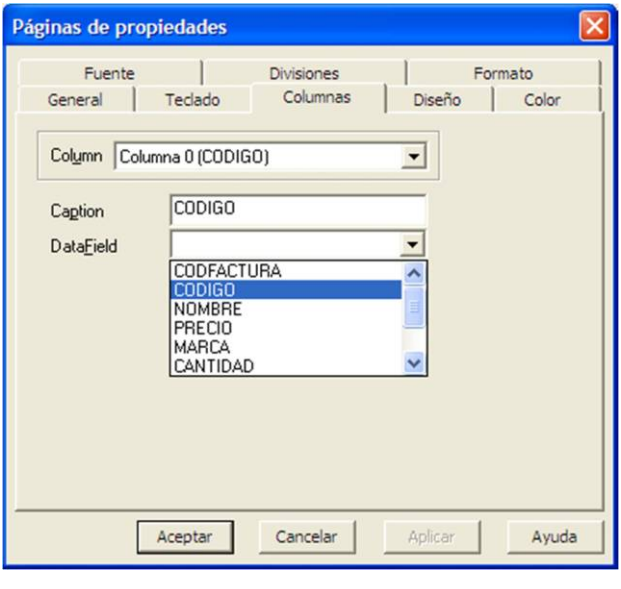

Damos click derecho sobre la grilla luego click en modificar luego volvemos a dar click derecho sobre la grilla luego agregar y lo haremos la cantidad de veces de acuerdo a los campos que necesitemos mostrar de la base de datos luego



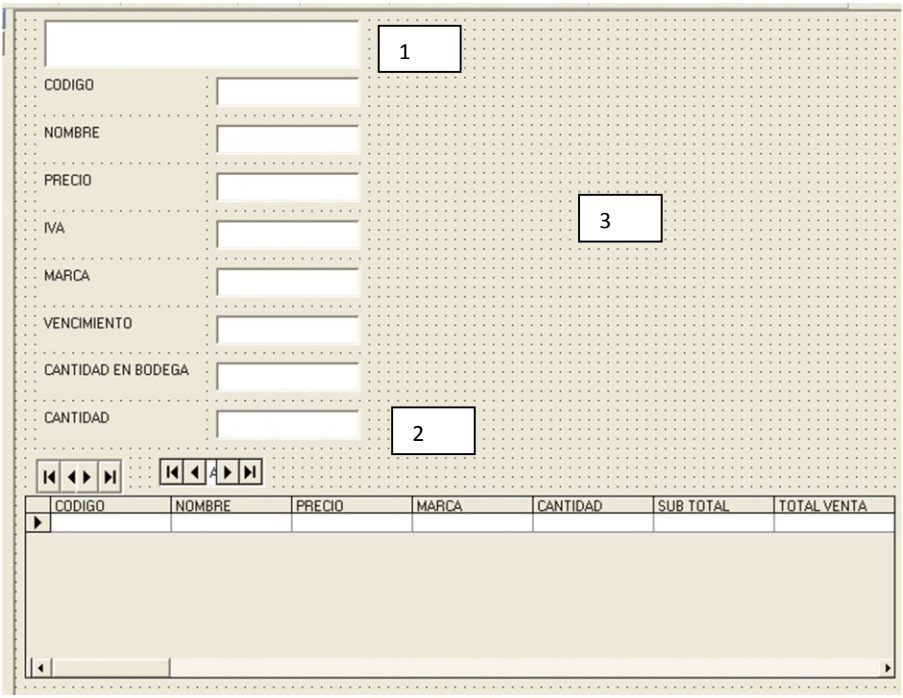

daremos click en propiedades para que nos salga esto




Voy a la pestaña columnas en **column** elijo la primera columna que inserte luego le doy un caption o nombre externo luego elijo el **datafield** que es el atributo de la entidad a la cual está conectado el adodc tal como muestra la siguiente imagen y así las veces que sea necesario de acuerdo a los paneles que se insertaron




A continuación veremos cómo queda el código fuente o programación de este ejemplo, **los números** que aparecen es para indicarle al usuario donde programar




Para programar damos doble click sobre la caja de texto del código en el evento keypress

```
Dim multiplicar1 as double
Dim total general as double   inicialización de variables
Dim total as double
Dim multiplicar as double
```

---

```
Private sub form_activate() 3
On error resume next
Iferr.number = 3021 then        activate significa en tiempo de activación
Endif                                   el adodc se mueva al último registro siempre  en caso de que se valla la luz continúe el programa en el ultimo valor guardado
Adodc1.recordset.movelast

End sub
```

---

```
Private sub text1_keypress(keyascii as integer) 1
If keyascii = 13 then
Data1.refresh
Data1.recordset.movefirst
Do while not data1.recordset.eof
If text1.text = data1.recordset.fields("cod articulo") then
Exit do
Else
Data1.recordset.movenext
End if
Loop
```

```
On error resume next
If err.number = 3021 then
End if
Text2.text = data1.recordset.fields("cod articulo")
Text3.text = data1.recordset.fields("nombre")
Text4.text = data1.recordset.fields("preciosalida")
Text5.text = data1.recordset.fields("iva")
Text6.text = data1.recordset.fields("marca")
Text7.text = data1.recordset.fields("fechavencimiento")
Text8.text = data1.recordset.fields("cantidad")
 On error resume next
If err.number = 3021 then
Endif
If text1.text <>data1.recordset.fields("cod articulo") then<> esto significa que si no
es igual al valor guardado en la base de datos
Msgbox "codigo no existe en la base de datos", vbcritical, "hipermarket"
Text1.setfocus
Text1.text = ""
Else
Text9.setfocus
Text9.selstart = len(text1)
Text1.text = ""
End if
End if
End sub
```

---

```
Private sub text9_change()    2
If val(text9.text) > 999999 then
Msgbox "error", vcritical, "cantidad"
Text9.text = ""
End if
End sub
```

---

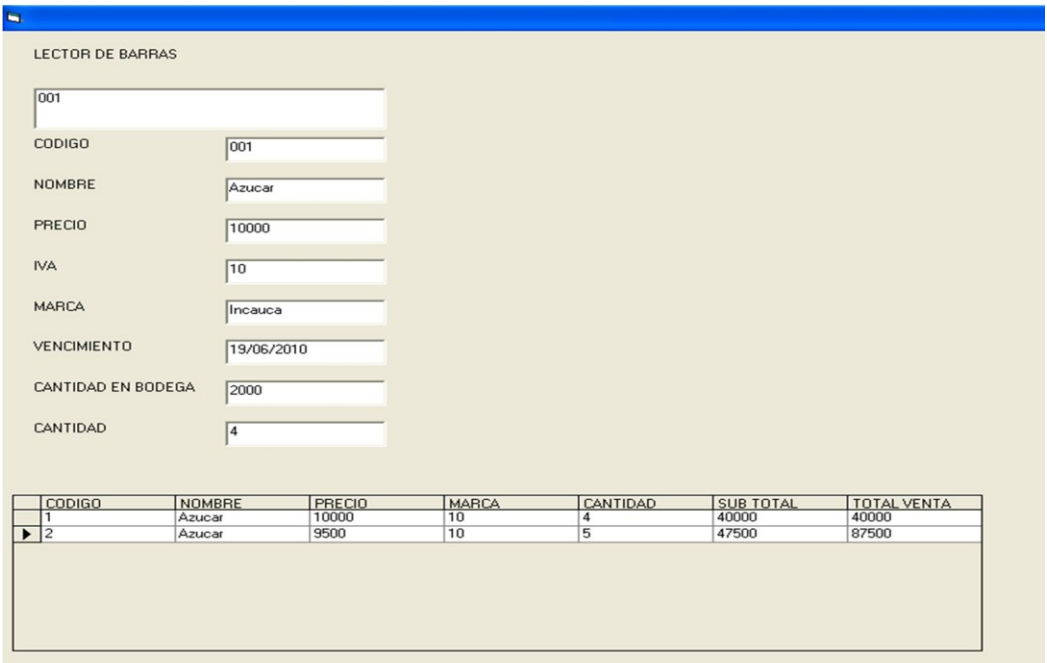
```
Private sub text9_keypress(keyascii as integer)  2
If keyascii = 13 then
If text9.text = "" or text9.text = 0 then
Msgbox "la cantidad no debe ser cero", vbcritical, "factura"
Text9.text = ""
Else
Adodc1.refresh
If adodc1.recordset.eof = true then
Multiplicar = val(text9.text) * val(text4.text)
Adodc1.recordset.addnew
Adodc1.recordset.fields("codigo") = val(text2.text)
Adodc1.recordset.fields("nombre") = text3.text
Adodc1.recordset.fields("precio") = val(text4.text)
```

```
Adodc1.recordset.fields("marca") = text5.text
Adodc1.recordset.fields("cantidad") = val(text9.text)
Adodc1.recordset.fields("subtotal") = multiplicar
Adodc1.recordset.fields("total") = multiplicar
Adodc1.recordset.update
Text2.text = ""
Text3.text = ""
Text4.text = ""
Text5.text = ""
Text6.text = ""
Text7.text = ""
Text8.text = ""
Text9.text = ""
Text1.setfocus
Else
Adodc1.refresh
Adodc1.recordset.movelast
Do while not adodc1.recordset.eof
If text2.text then
Exit do
Exit sub
Else
Adodc1.recordset.movenext
End if
Loop
On error resume next
If err.number = 3021 then
Msgbox "no deje la cajavacia", vbinformation, "softmarket"
End if
Multiplicar1 = val(text9.text) * val(text4.text)
Totalgeneral = multiplicar1 + adodc1.recordset.fields("total")
Adodc1.recordset.addnew
Adodc1.recordset.fields("codigo") = text2.text
Adodc1.recordset.fields("nombre") = text3.text
Adodc1.recordset.fields("precio") = text4.text
Adodc1.recordset.fields("marca") = text5.text
Adodc1.recordset.fields("cantidad") = text9.text
Adodc1.recordset.fields("subtotal") = multiplicar1
Adodc1.recordset.fields("total") = totalgeneral
Adodc1.recordset.update
Text2.text = ""
Text3.text = ""
Text4.text = ""
Text5.text = ""
Text6.text = ""
Text7.text = ""
Text8.text = ""
```

```
Text9.text = ""
Text1.setfocus
End if
End if
End if
End sub
```

---

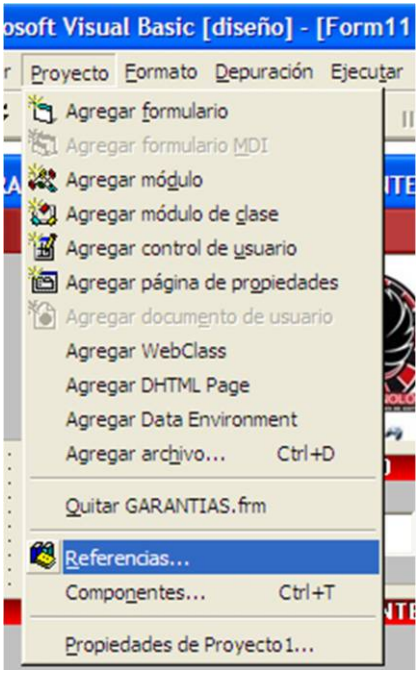
En tiempo de ejecución quedara así



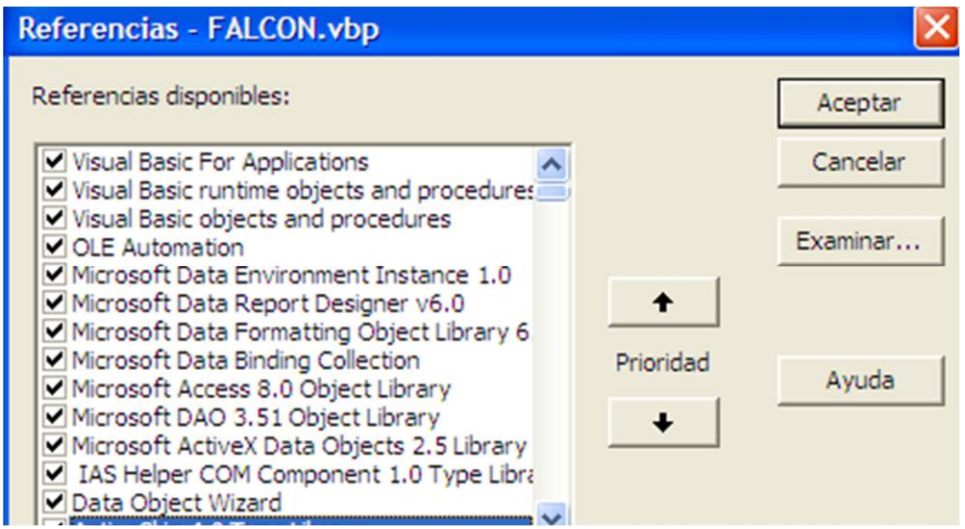

## 28. BORRADO GENERAL DE DATOS, MACROS E IMPRESIÓN DE INFORMES

A continuación borrado general de datos de Access macros en Access para crear los botones de imprimir los informes de Access e imprimir desde visual

Para este ejemplo lo primero que debo hacer es lo siguiente

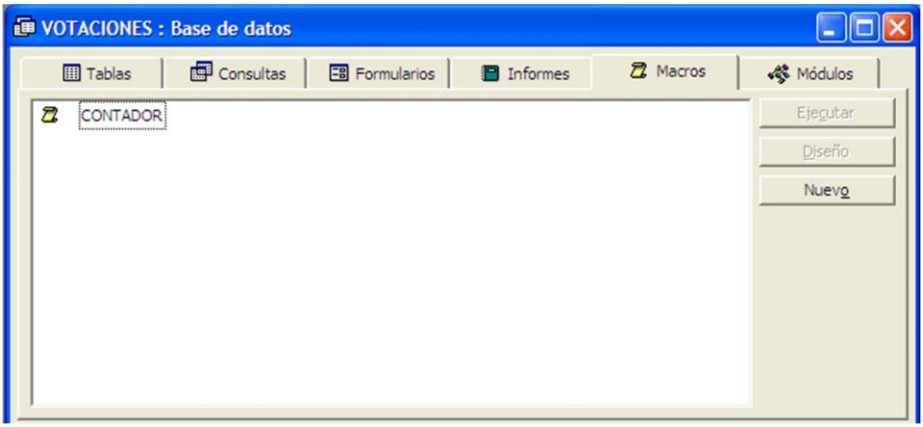



Click en el menú proyecto referencias y que queden chuleadas las casillas de verificación de la siguiente manera y opciones



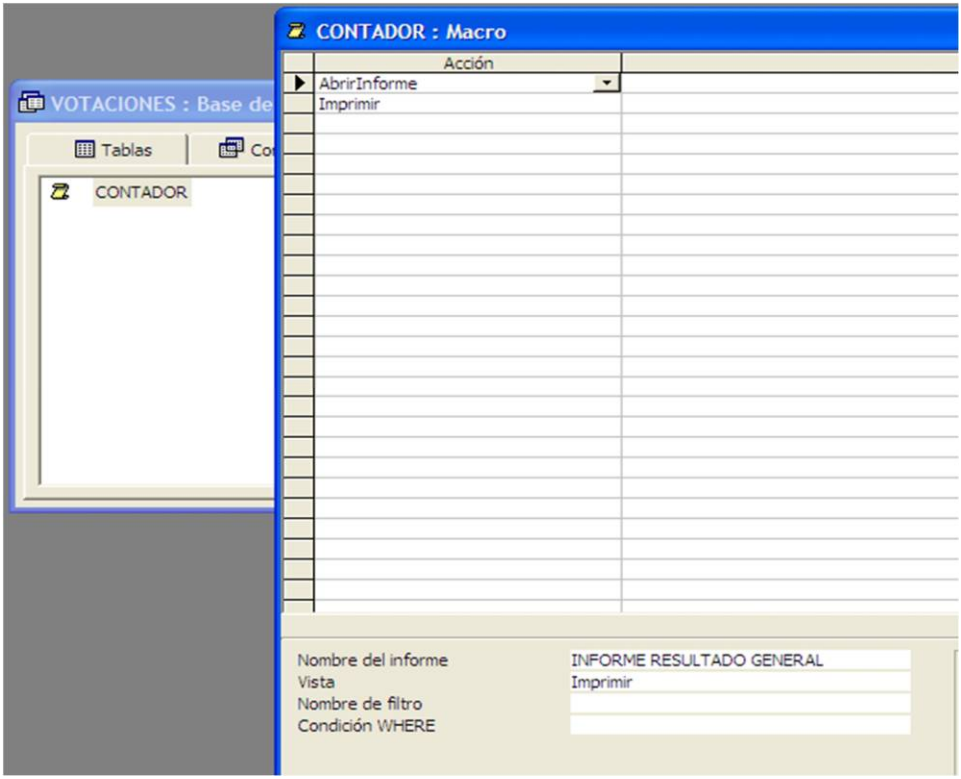

Esto para que visual reconozca los comandos de Access

A continuación botón de borrado general de datos de Access desde imprimir un informe un botón pero antes explicaremos como crear una macro en Access obviamente ya debe estar hecho el informe

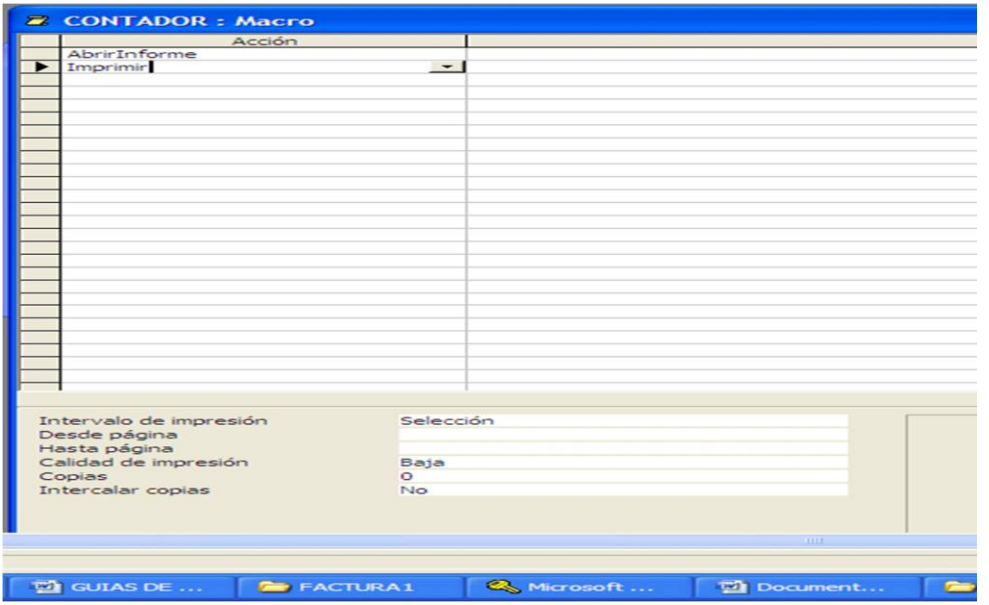


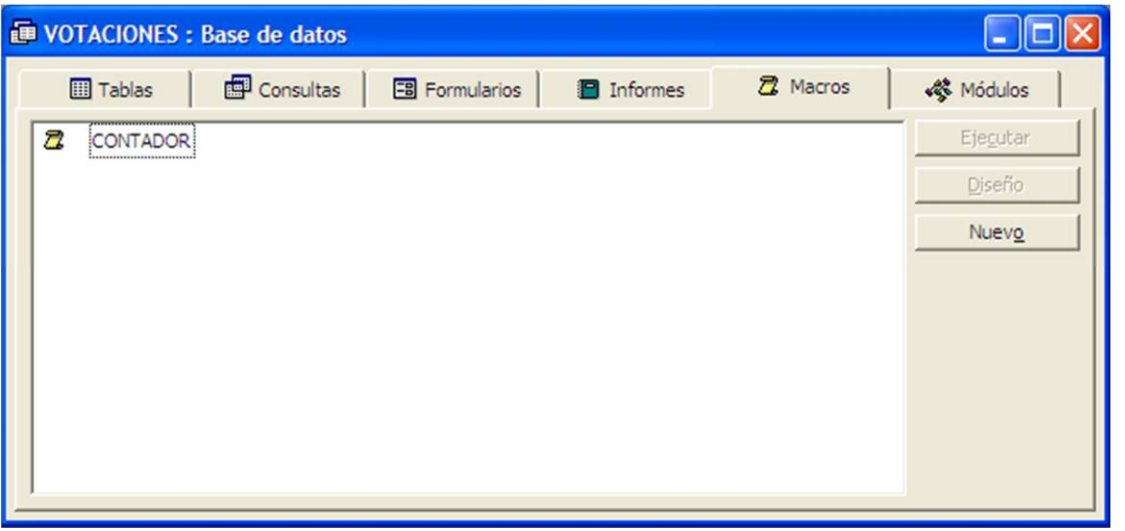

Damos click en la pestaña macros luego en el botón nuevo sale esto este es el diseño para imprimir

Luego damos click en la x roja que aparece arriba para guardar y guardamos la

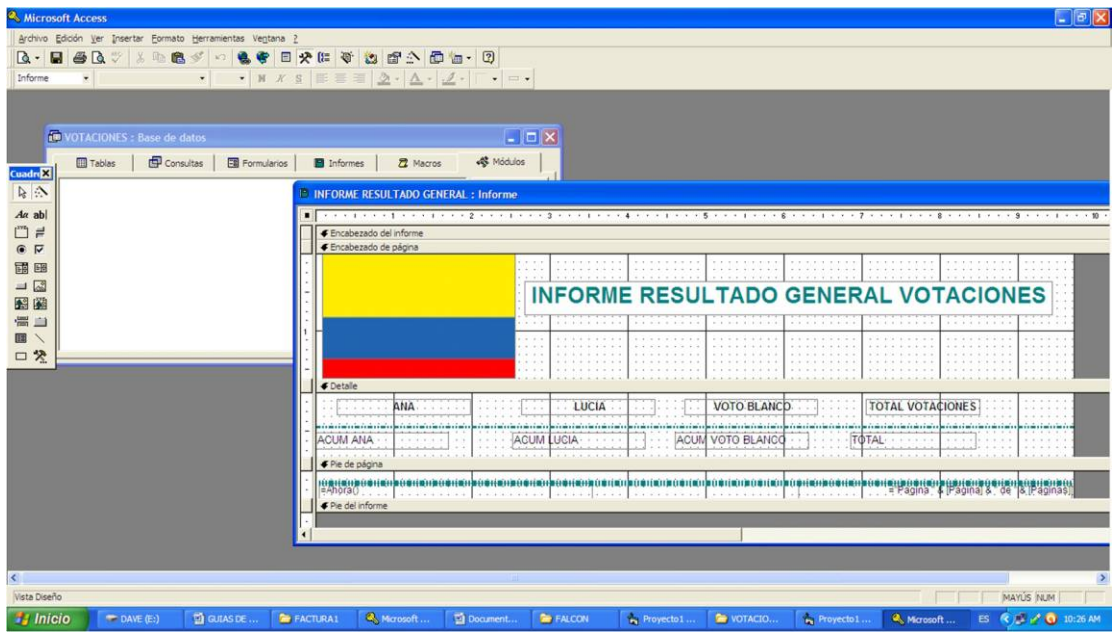


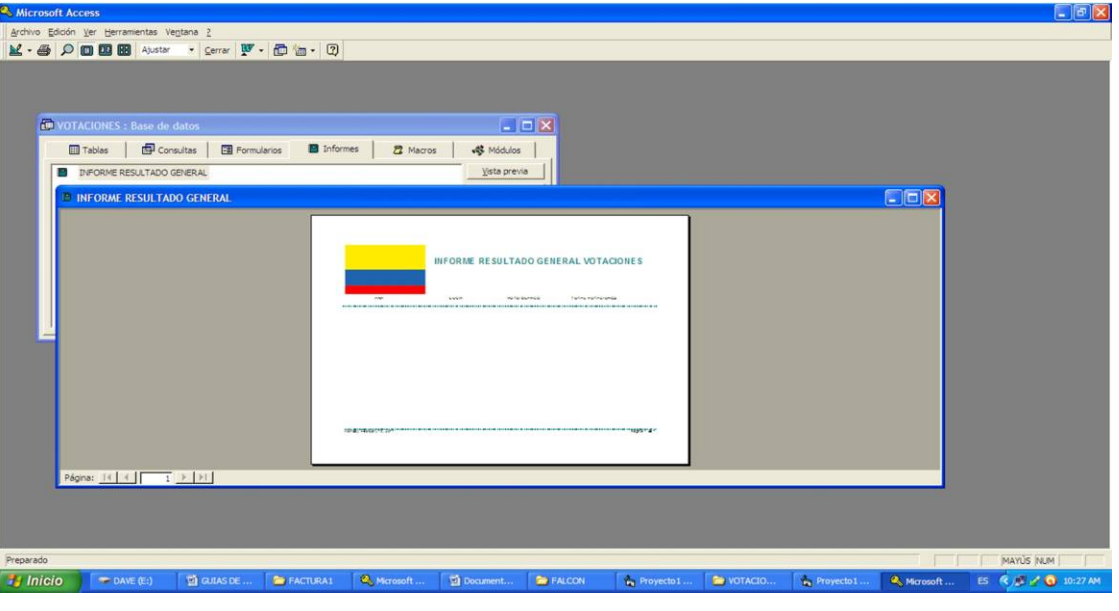

macro con un nombre quedara guardada así




Esta macro va imprimir el siguiente informe


Microsoft Access
Archivo Edición Ver Insertar Formato Herramientas Ventana ?
VOTACIONES : Base de datos
Tablas Consultas Formularios Informes Macros Módulos
INFORME RESULTADO GENERAL : Informe
Encabezado del informe
Encabezado de página
INFORME RESULTADO GENERAL VOTACIONES
Detalle
ANA
LUCIA
VOTO BLANCO
TOTAL VOTACIONES
ACUM ANA
ACUM LUCIA
ACUM VOTO BLANCO
TOTAL
Pie de página
Ahora()
Página & [Página] & de & [Páginas]
Pie del informe
Vista Diseño
MAYÚS NUM
Inicio
10:26 AM


Microsoft Access
Archivo Edición Ver Herramientas Ventana ?
Ajustar
Cerrar
VOTACIONES : Base de datos
Tablas Consultas Formularios Informes Macros Módulos
INFORME RESULTADO GENERAL
Vista previa
INFORME RESULTADO GENERAL
INFORME RESULTADO GENERAL VOTACIONES
Página:
Preparado
MAYÚS NUM
Inicio
10:27 AM

A continuación la programación del botón imprimir y borrado general de datos de Access

**La programación de este botón queda así**

```
Dim w as new access.application
W.opencurrentdatabase ("c:\archivos de
programa\votaciones\votaciones.mdb")
On error resume next
If err.number = 2200 then
End if
W.docmd.runmacro "contador"
```

Esta instrucción es la que permite imprimir la macro de Access

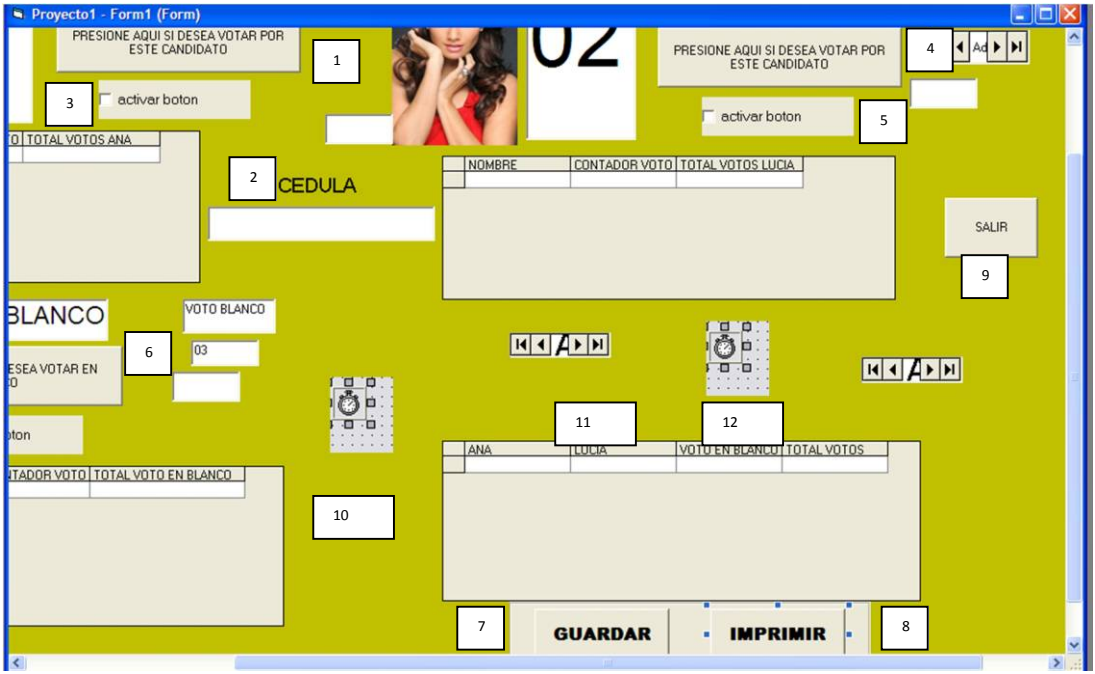
**W** es una variable cualquiera de tipo Access para la ejecución de la macro, **opencurrentdatabase** es la ruta donde se encuentra la base de datos

**On error** es para obviar un error que se presenta al ejecutar la macro **y docmd.runmacro** es para que ejecute la macro a continuación la programación de borrado general de datos de Access

```
Set c = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
C.execute ("delete from contador")
Adodc4.refresh
W.closecurrentdatabase
End sub
```

**Delete from "contador"** significa que borre todos los datos que están en esta entidad o tabla  a continuación un ejemplo completo de una aplicación con acumulados impresiones guardados etc., recuerde que los números que aparecen solo es para indicar al usuario donde programar

Se utilizaron adodcs cajas de texto labels con programación de marquesina, checkbox o casillas de verificación command button




La programación de todo este proyecto es el siguiente

Recuerden que la variables se inicializan en

Dim acumlucia, acumana, acumblanco, contlucia, contana, contblanco, total as double
Dim a as database
Dim b as database
Dim v as database
Dim c as database
Dim f as database
Dim strtexto as string
Dim strtexto1 as string

```
Dim accion1 as boolean
Dim accion as boolean
```

---

```
Private sub check1_click()
Command1.enabled = true
Text13.setfocus
End sub
```

---

```
Private sub check2_click()
Command2.enabled = true
Text13.setfocus
End sub
```

---

```
Private sub check3_click() 3
Command3.enabled = true
Text13.setfocus
End sub
```

---

```
Private sub command1_click() 1
If text13.text = "" then
Msgbox "debe ingresar su numero de cedula", vbinformation, "registraduria"
Else
Adodc1.refresh
If adodc1.recordset.eof = true then
Contana = 1
Acumana = contana
Adodc1.recordset.addnew
Adodc1.recordset.fields("cod candidato") = val(text8.text)
Adodc1.recordset.fields("nombre") = text4.text
Adodc1.recordset.fields("cedula del sufragante") = text13.text
Adodc1.recordset.fields("contador ana") = contana
Adodc1.recordset.fields("acum ana") = acumana
Adodc1.recordset.update
Text13.text = ""
Msgbox "su voto se a contabilizado satisfactoriamente", vbinformation,
"registraduria"
Command1.enabled = false
Else
Adodc1.refresh
Adodc1.recordset.movelast
Do while not adodc1.recordset.eof
If text8.text then
Exit do
Else
Adodc1.recordset.movenext
End if
Loop
```

```
On error resume next
If err.number = -2147467259 then
Msgbox "voto no se puede contabilizar porque el sufragante con este numero de cedula ya a contabilizado su voto", vbcritical, "registraduria"
End if
If text13.text = adodc1.recordset.fields("cedula del sufragante") then
Msgbox "voto no se puede contabilizar porque el sufragante con este numero de cedula ya a contabilizado su voto", vbcritical, "registraduria"
Else
Contana = 1
Acumana = adodc1.recordset.fields("acum ana") + contana
Adodc1.recordset.addnew
Adodc1.recordset.fields("cod candidato") = text8.text
Adodc1.recordset.fields("nombre") = text4.text
Adodc1.recordset.fields("contador ana") = contana
Adodc1.recordset.fields("cedula del sufragante") = text13.text
Adodc1.recordset.fields("acum ana") = acumana
Adodc1.recordset.update
Msgbox "su voto se a contabilizado satisfactoriamente", vbinformation, "registraduria"
Command1.enabled = false
Text13.text = ""
End if
End if
End if
End sub
```

---

```
Private sub command2_click()   4
If text13.text = "" then
Msgbox "debe ingresar su numero de cedula", vbinformation, "registraduria"
Else
Adodc2.refresh
If adodc2.recordset.eof = true then
Contlucia = 1
Acumlucia = contlucia
Adodc2.recordset.addnew
Adodc2.recordset.fields("cod candidato") = text7.text
Adodc2.recordset.fields("nombre") = text5.text
Adodc2.recordset.fields("cedula del sufragante") = text13.text
Adodc2.recordset.fields("contador lucia") = contlucia
Adodc2.recordset.fields("acum lucia") = acumlucia
Adodc2.recordset.update
Msgbox "su voto se a contabilizado satisfactoriamente", vbinformation, "registraduria"
Command2.enabled = false
Text13.text = ""
Else
Adodc2.refresh
```

```
Adodc2.recordset.movelast
Do while not adodc2.recordset.eof
If text7.text then
Exit do
Else
Adodc2.recordset.movenext
End if
Loop
On error resume next
If err.number = -2147467259 then
Msgbox "voto no se puede contabilizar porque el sufragante con este numero de cedula ya a contabilizado su voto", vbcritical, "registraduria"
End if
If text13.text = adodc2.recordset.fields("cedula del sufragante") then
Msgbox "voto no se puede contabilizar porque el sufragante con este numero de cedula ya a contabilizado su voto", vbcritical, "registraduria"
Else
Contlucia = 1
Acumlucia = adodc2.recordset.fields("acum lucia") + contlucia
Adodc2.recordset.addnew
Adodc2.recordset.fields("cod candidato") = text7.text
Adodc2.recordset.fields("nombre") = text5.text
Adodc2.recordset.fields("contador lucia") = contlucia
Adodc2.recordset.fields("cedula del sufragante") = text13.text
Adodc2.recordset.fields("acum lucia") = acumlucia
Adodc2.recordset.update
Msgbox "su voto se a contabilizado satisfactoriamente", vbinformation, "registraduria"
Command2.enabled = false
Text13.text = ""
End if
End if
End if
End sub
```

---

```
Private sub command3_click()  6
If text13.text = "" then
Msgbox "debe in gresar su numero de cedula", vbinformation, "registraduria"
Else
Adodc3.refresh
If adodc3.recordset.eof = true then
Contblanco = 1
Acumblanco = contblanco
Adodc3.recordset.addnew
Adodc3.recordset.fields("cod voto") = text9.text
Adodc3.recordset.fields("cedula del sufragante") = text13.text
Adodc3.recordset.fields("nombre") = text6.text
```

```
Adodc3.recordset.fields("contador voto blanco") = contblanco
Adodc3.recordset.fields("acum voto blanco") = acumblanco
Adodc3.recordset.update
Msgbox "su voto se a contabilizado satisfactoriamente", vbinformation, "registraduria"
Command3.enabled = false
 Text13.text = ""
Else
Adodc3.refresh
Adodc3.recordset.movelast
Do while not adodc3.recordset.eof
If text9.text then
Exit do
Else
Adodc3.recordset.movenext
End if
Loop
On error resume next
If err.number = -2147467259 then
Msgbox "voto no se puede contabilizar porque el sufragante con este numero de cedula ya a contabilizado su voto", vbcritical, "registraduria"
End if
If text13.text = adodc3.recordset.fields("cedula del sufragante") then
Msgbox "voto no se puede contabilizar porque el sufragante con este numero de cedula ya a contabilizado su voto", vbcritical, "registraduria"
Else
Contblanco = 1
Acumblanco = adodc3.recordset.fields("acum voto blanco") + contblanco
Adodc3.recordset.addnew
Adodc3.recordset.fields("cod voto") = text9.text
Adodc3.recordset.fields("nombre") = text6.text
Adodc3.recordset.fields("contador voto blanco") = contblanco
Adodc3.recordset.fields("cedula del sufragante") = text13.text
Adodc3.recordset.fields("acum voto blanco") = acumblanco
Adodc3.recordset.update
Msgbox "su voto se a contabilizado satisfactoriamente", vbinformation, "registraduria"
Command3.enabled = false
Text13.text = ""
End if
End if
End if
End sub
```

---

```
Private sub command4_click() 9
End
End sub
```

```
Private sub command5_click()  7
Adodc4.refresh
Adodc5.refresh
Total = val(text10.text) + val(text11.text) + val(text12.text)
Adodc5.recordset.addnew
Adodc4.recordset.addnew
Adodc4.recordset.fields("acum ana") = val(text10.text)
Adodc4.recordset.fields("nombre") = text4.text
Adodc5.recordset.fields("acum ana") = val(text10.text)
Adodc5.recordset.fields("nombre") = text4.text
Adodc5.recordset.fields("nombre") = text5.text
Adodc5.recordset.fields("nombre") = text6.text
Adodc5.recordset.fields("acum lucia") = val(text11.text)
Adodc5.recordset.fields("acum voto blanco") = val(text12.text)
Adodc5.recordset.fields("total") = total
Adodc4.recordset.fields("nombre") = text5.text
Adodc4.recordset.fields("nombre") = text6.text
Adodc4.recordset.fields("acum lucia") = val(text11.text)
Adodc4.recordset.fields("acum voto blanco") = val(text12.text)
Adodc4.recordset.fields("total") = total
Adodc5.recordset.update
Adodc4.recordset.update
Msgbox "su voto se a contabilizado satisfactoriamente", vbinformation, "registraduria"
Ifadodc4.recordset.fields("acum ana") > adodc4.recordset.fields("acum lucia") and adodc4.recordset.fields("acum ana") > adodc4.recordset.fields("acum voto blanco") then
Msgbox "ana has ganado en esta mesa", vbinformation, "registraduria"
Set a = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
A.execute ("delete from candidato1")
Set b = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
B.execute ("delete from candidato2")
Set v = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
V.execute ("delete from voto")
Adodc1.refresh
Adodc2.refresh
Adodc3.refresh
Datagrid3.refresh
End if
If adodc4.recordset.fields("acum lucia") > adodc4.recordset.fields("acum ana") and adodc4.recordset.fields("acum lucia") > adodc4.recordset.fields("acum voto blanco") then
Msgbox "lucia has ganado en esta mesa", vbinformation, "registraduria"
Set a = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
A.execute ("delete from candidato1")
Set b = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
```

```
B.execute ("delete from candidato2")
Set v = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
V.execute ("delete from voto")
Adodc1.refresh
Adodc2.refresh
Adodc3.refresh
Datagrid3.refresh
End if
If adodc4.recordset.fields("acum voto blanco") > adodc4.recordset.fields("acum lucia") and adodc4.recordset.fields("acum voto blanco") > adodc4.recordset.fields("acum ana") then
Msgbox "voto en blanco supera mcandidatos en esta mesa", vbinformation, "registraduria"
Set a = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
A.execute ("delete from candidato1")
Set b = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
B.execute ("delete from candidato2")
Set v = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
V.execute ("delete from voto")
Adodc1.refresh
Adodc2.refresh
Adodc3.refresh
Datagrid3.refresh
End if
If adodc4.recordset.fields("acum ana") = adodc4.recordset.fields("acum lucia") or adodc4.recordset.fields("acum ana") = adodc4.recordset.fields("acum voto blanco") or adodc4.recordset.fields("acum lucia") = adodc4.recordset.fields("acum voto blanco") or adodc4.recordset.fields("acum lucia") = adodc4.recordset.fields("acum voto blanco") = adodc4.recordset.fields("acum ana") then
Msgbox "han empatado no ha ganado nadie", vbinformation, "registraduria"
Set a = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
A.execute ("delete from candidato1")
Set b = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
B.execute ("delete from candidato2")
Set v = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
V.execute ("delete from voto")
Adodc1.refresh
Adodc2.refresh
Adodc3.refresh
Datagrid3.refresh
End if
End sub
Private sub command6_click()
Dim w as new access.application
W.opencurrentdatabase ("c:\archivos de programa\votaciones\votaciones.mdb")
On error resume next
If err.number = 2200 then
```

```
End if
W.docmd.runmacro "contador"
Set c = opendatabase("c:\archivos de programa\votaciones\votaciones.mdb")
C.execute ("delete from contador")
Adodc4.refresh
W.closecurrentdatabase
End sub
```

---

```
Private sub form_activate() 10
On error resume next
If err.number = 3021 then
End if
Adodc1.recordset.movelast
Adodc2.recordset.movelast
Adodc3.recordset.movelast
End sub
```

---

```
Private sub form_load()  10
Strtexto = string(5, " ") + string(5, " ") + "bienvenidos a votaciones de alcalde"
Accion = true
Strtexto1 = string(5, " ") + string(5, " ") + "resultados generales"
Accion1 = true
Form1.height = 10335   esto es para que el formulario quede igual de tamaño en tiempo de ejecución en caso que lo modifique al programarlo esto está en la barra de propiedades de formulario
Form1.width = 19320
End sub
```

---

```
Private sub timer1_timer() 11
Strtexto = mid(strtexto, 2) & left(strtexto, 1)
Label1.caption = strtexto
End sub
```

---

```
Private sub timer2_timer() 12
Strtexto1 = mid(strtexto1, 2) & left(strtexto1, 1)
Label2.caption = strtexto1
End sub
```

---

# SEGUNDO EJEMPLO DE ACUMULADOS Y MACROS EN ACCES

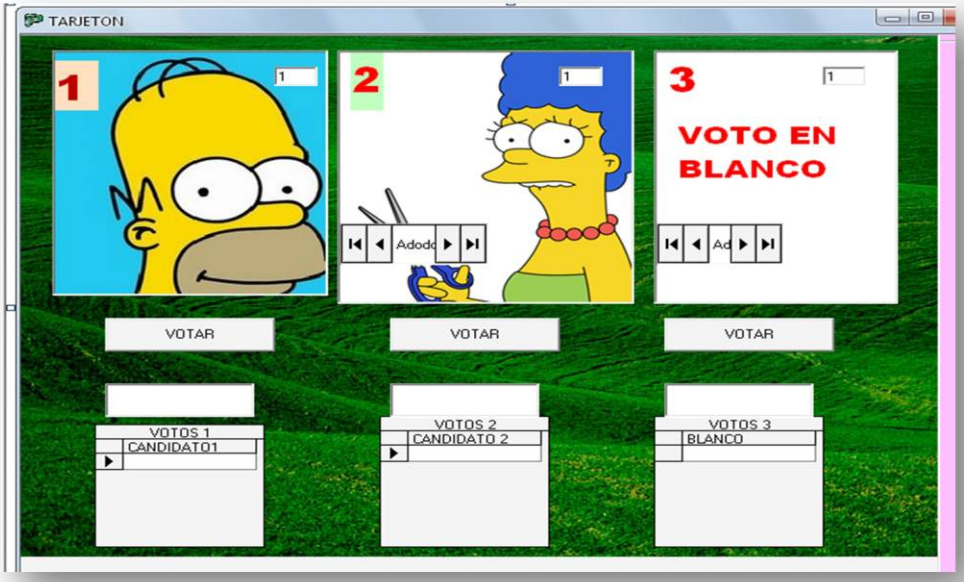
A continuación mostraremos la pantalla de bienvenida

A continuación mostrare el contabilizador de cédulas donde impide que una se vuelva a repetir y pueda votar más veces

El código fuente del siguiente cuadro de dialogo queda asi

```
EDULA                                                    KeyPress
Private Sub Command1_Click()
CONTRASEÑA.Show
End Sub
Private Sub Form_Load()
Unload frmSplash
End Sub

Private Sub TXCEDULA_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
Adodc1.Refresh
On Error Resume Next
If Err.Number = 3021 Then
End If
Adodc1.Recordset.MoveFirst
Do While Not Adodc1.Recordset.EOF
If Adodc1.Recordset.Fields("CEDULA") <> TXCEDULA.Text Then
Adodc1.Recordset.MoveNext
Else
Exit Do
End If
Loop
If Adodc1.Recordset.EOF Then
Adodc1.Refresh
Adodc1.Recordset.AddNew
Adodc1.Recordset.Fields("CEDULA") = Val(TXCEDULA.Text)
Adodc1.Recordset.Update
MsgBox "SU CEDULA SE A CONTABILIZADO SATISFACTORIAMENTE", vbInformation, "VOTACION YO"
TXCEDULA = ""
Form2.Show
Else
MsgBox "Este registro se encuentra en la base de datos, digite otra cedula", vbInformation, "Error"
End If
End If
End Sub
```

Si la cédula no había sido contabilizada ingresara a la siguiente opción para votar



El código fuente de la siguiente cuadro de dialogo es

```
Private Sub Command1_Click()
Txsuma1.Text = Val(Txsuma1.Text) + Val(CONT1.Text)
Adodc1.Refresh
Adodc1.Recordset.AddNew
Adodc1.Recordset.Fields("VotoPri1") = Val(CONT1.Text)
Adodc1.Recordset.Fields("Suma") = Val(Txsuma1.Text)
Adodc1.Recordset.Update
Unload Me
Form1.Show
Unload Me
End Sub
Private Sub Command2_Click()
Txsuma3.Text = Val(Txsuma3.Text) + Val(CONT3.Text)
Adodc3.Refresh
Adodc3.Recordset.AddNew
Adodc3.Recordset.Fields("BLANCO") = Val(CONT3.Text)
Adodc3.Recordset.Fields("SUMA") = Val(Txsuma3.Text)
Adodc3.Recordset.Update
Unload Me
Form1.Show
Unload Me
End Sub
Private Sub Command3_Click()
Txsuma2.Text = Val(Txsuma2.Text) + Val(CONT2.Text)
Adodc2.Refresh
```

```
Adodc2.Recordset.AddNew

Adodc2.Recordset.Fields("VotoPri2") = Val(CONT2.Text)

Adodc2.Recordset.Fields("SUMA") = Val(Txsuma2.Text)

Adodc2.Recordset.Update

Unload Me

Form1.Show

Unload Me

End Sub

Private Sub Form_Load()

Unload Form1

Adodc1.Refresh

On Error Resume Next

If Err.Number = 3021 Then

End If

Adodc1.Recordset.MoveLast

Txsuma1.Text = Adodc1.Recordset.Fields("Suma")

Adodc2.Refresh

Adodc2.Recordset.MoveLast

Txsuma2.Text = Adodc2.Recordset.Fields("SUMA")

Adodc3.Refresh

Adodc3.Recordset.MoveLast

Txsuma3.Text = Adodc3.Recordset.Fields("SUMA")

End Sub
```

Para cerrar votaciones será este cuadro de dialogo

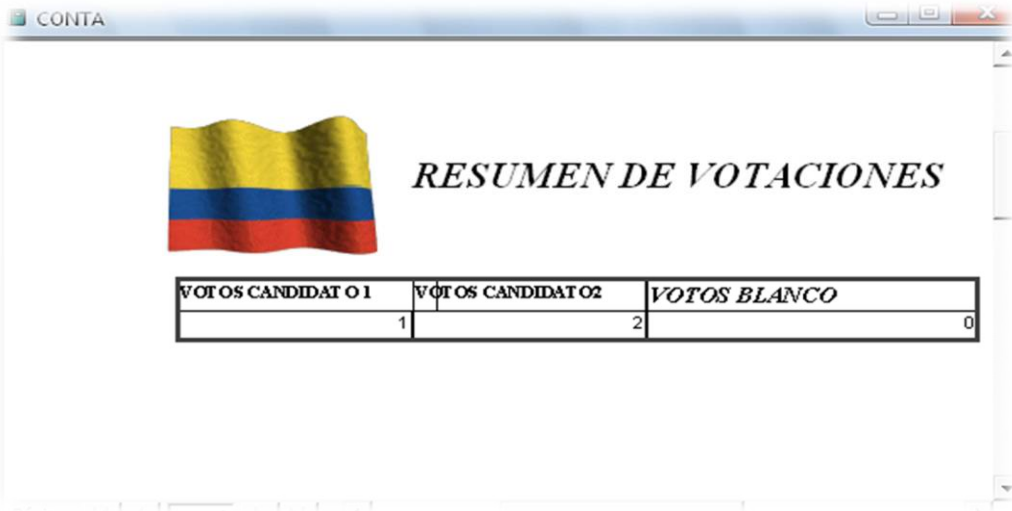
El código fuente de el siguiente cuadro de dialogo es

```
Dim b As Database

Dim d As Database

Dim c As Database

Dim A As Database

Private Sub CIERRE_Click()

Data4.Refresh

Data4.Recordset.AddNew

On Error Resume Next

If Err.Number = 3421 Then

End If

Data4.Recordset.Fields("VOTOS PRINCESA 1") = TXPRIN1.Text

Data4.Recordset.Fields("VOTOS PRINCESA 2") = TXPRIN2.Text

Data4.Recordset.Fields("VOTOS BLANCO") = TXBLANCO.Text

Data4.Recordset.Update

If TXPRIN1.Text > TXPRIN2.Text And TXPRIN1.Text > TXBLANCO.Text Then
```

```
MsgBox "A GANADO CANDIDATO # 1", vbInformation, "VOTACION"

End If

If TXPRIN2.Text > TXPRIN1.Text And TXPRIN2.Text > TXBLANCO.Text Then

MsgBox "A GANADO EL CANDIDATO # 2", vbInformation, "VOTACION"

End If

If TXBLANCO.Text > TXPRIN1.Text And TXBLANCO.Text > TXPRIN2.Text Then

MsgBox "A GANADO VOTO EN BLANCO # 1", vbInformation, "VOTACION"

End If

If TXBLANCO.Text = TXPRIN1.Text Or TXBLANCO.Text = TXPRIN2.Text And
TXPRIN2.Text = TXPRIN1.Text Then

MsgBox "HAN QUEDADO EN EMPATE", vbInformation, "VOTACION"

End If

CIERRE.Enabled = False

End Sub

Private Sub Command1_Click()

End

End Sub

Private Sub Command2_Click()

Set A = OpenDatabase("C:\Archivos de programa\VOTACIONES YO\CEDULAS.mdb")

A.Execute ("delete FROM  PRINCESA")

Set b = OpenDatabase("C:\Archivos de programa\VOTACIONES YO\CEDULAS.mdb")

b.Execute ("delete FROM  PRINCESA2")

Set c = OpenDatabase("C:\Archivos de programa\VOTACIONES YO\CEDULAS.mdb")

c.Execute ("delete FROM  BLANCO")

Set d = OpenDatabase("C:\Archivos de programa\VOTACIONES YO\CEDULAS.mdb")

d.Execute ("delete FROM  CEDULA")

Dim w As New Access.Application
```

```
w.OpenCurrentDatabase ("C:\Archivos de programa\VOTACIONES YO\CEDULAS.mdb")

On Error Resume Next

If Err.Number = 2200 Then

End If

w.DoCmd.RunMacro "MACRO"

Command2.Enabled = False

End Sub

Private Sub Form_Load()

Data1.Refresh

On Error Resume Next

If Err.Number = 3021 Then

End If

Data1.Recordset.MoveLast

TXPRIN1.Text = Data1.Recordset.Fields("Suma")

Data2.Refresh

Data2.Recordset.MoveLast

TXPRIN2.Text = Data2.Recordset.Fields("SUMA")

Data3.Refresh

Data3.Recordset.MoveLast

TXBLANCO.Text = Data3.Recordset.Fields("SUMA")

End Sub
```

Al final si deseo imprimir el informe se vera así



# 29. SPLASH O PANTALLA DE BIENVENIDA

A continuación mostraremos como crear el **splash** o pantalla de bienvenida y como iniciar al programa con contraseñas

Para sacar la pantalla **de bienvenida o splash** haremos esto

Damos click en el icono que indica la flecha luego seleccionamos pantalla de inicio y abrir

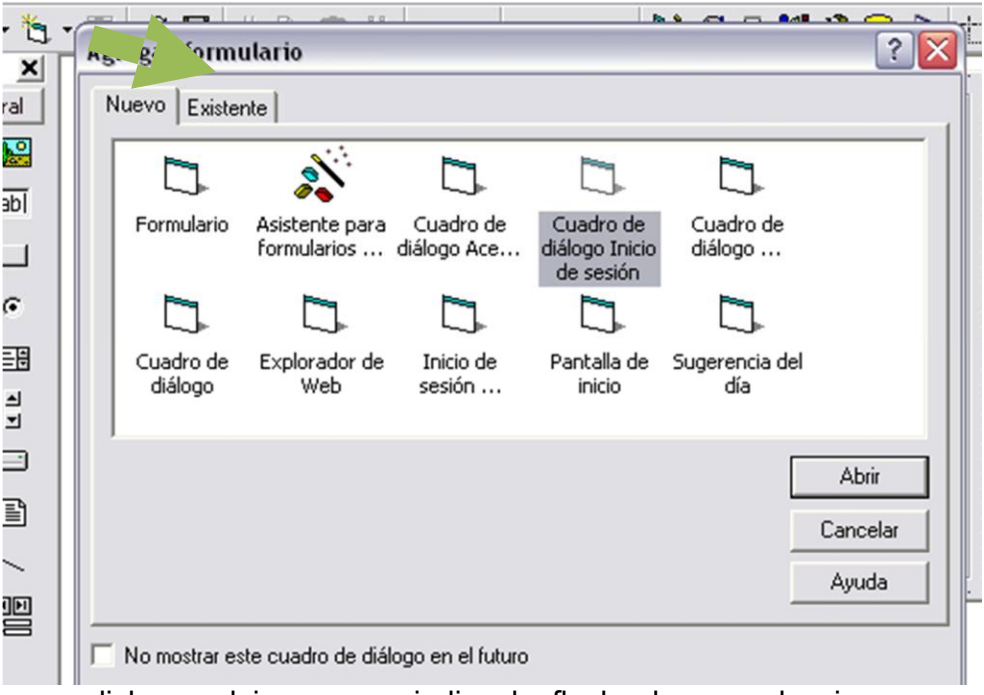
Damos doble click y borramos la programación que trae por defecto luego empiezo a modificar el fondo la letra **el caption** le cargo otra imagen en el

**picture** tal como muestro a continuación

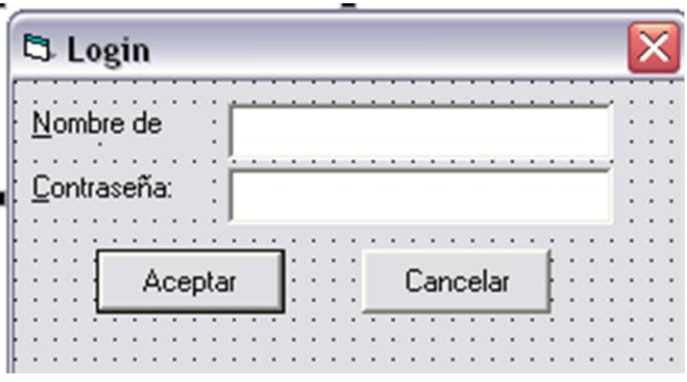
Click en la propiedad picture y cargar la imagen en la carpeta del programa y abrir después de modificar todo el splash modelo quedara así

# 30. INCIO DE SECION AL SOFTWARE

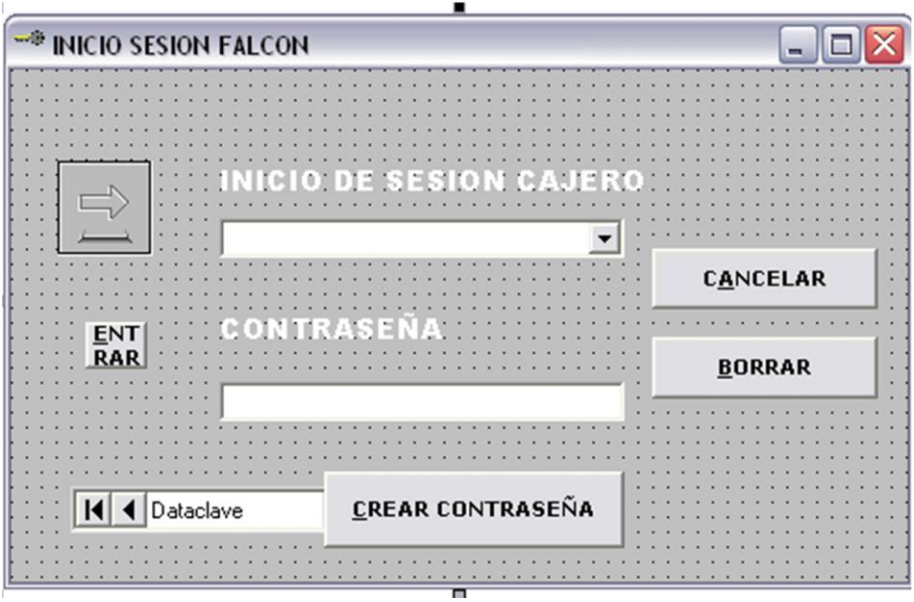
Para sacar la pantalla de **inicio de sesión** es así




Damos click en el icono que indica la flecha luego seleccionamos cuadro de dialogo de inicio de sesión y abrir quedara así

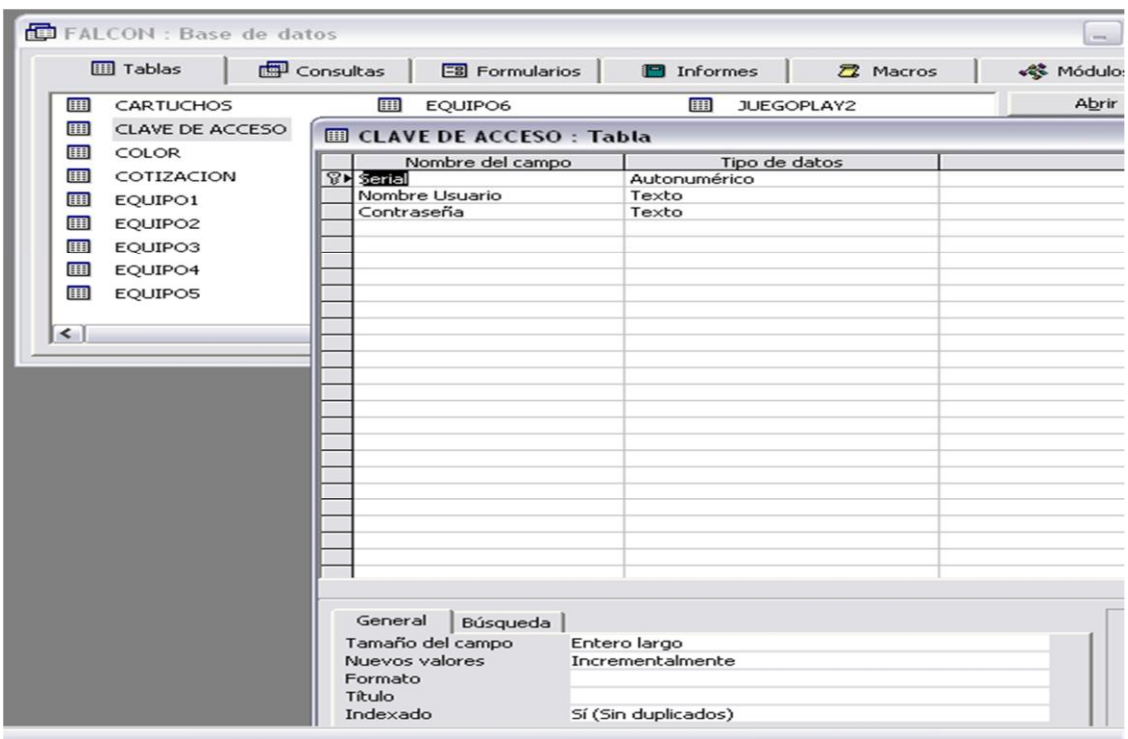



Este cuadro también tiene una programación la borramos y modificamos este formulario de la siguiente manera

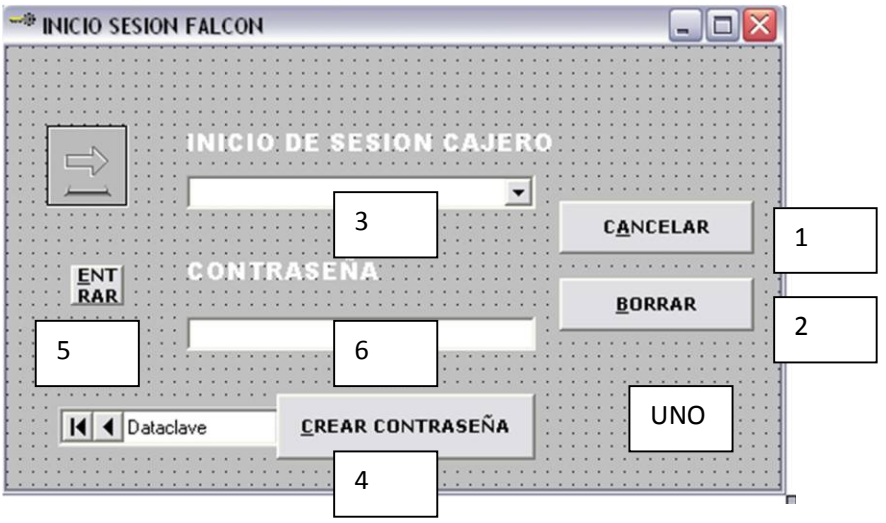



Le cargamos un data conectado con una entidad de la base de datos del programa un **dbcombo** y conectarlo con el data y la entidad tal como explique antes cajas de texto labels entre otros a continuación veremos cómo queda la programación

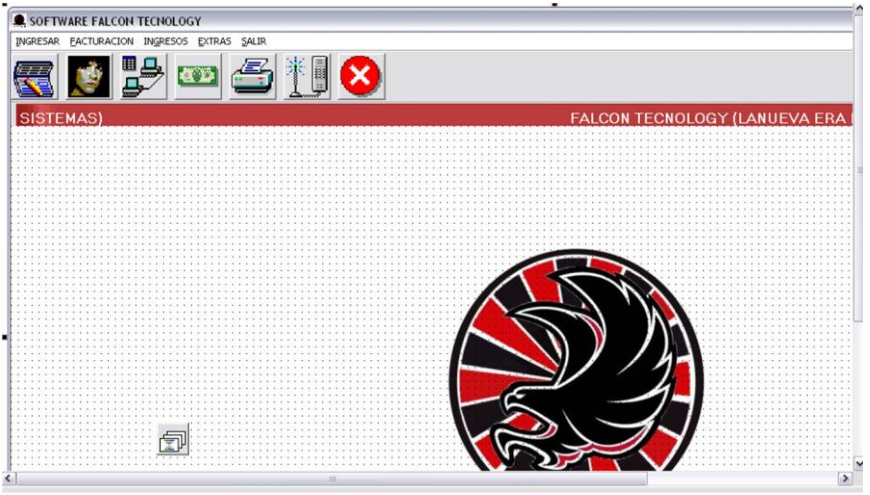
Para ello crear una tabla en Access con estas especificaciones y conectar el data a esta entidad tal como se explicó anteriormente




A continuación muestro imágenes del proyecto y después toda la programación se necesita tener el mdi la pantalla de inicio y la pantalla de inicio de sesión para este proyecto




A continuación mostrare la programación de acuerdo al orden de las imágenes




Esta es la programación de cada control de la imagen uno

```
Dim c as integer
Dim bandera as variant variable que sirve para cualquier tipo de datos
```

---

```
Private sub cancelbutton_click() 1
Unload me
End sub
```

---

```
Private sub command1_click() 4
Dataclave.refresh
Dataclave.recordset.addnew
Dataclave.recordset.fields("nombreusuario") = dbcusuario.text
Dataclave.recordset.fields("contraseña") = txclave.text
Dataclave.recordset.update
Msgbox "su contraseña ha sido creada satisfactoriamente", vbinformation, "softmarket"
Dbcusuario.text = ""
Txclave.text = ""
End sub
Private sub command2_click() 5
If dbcusuario.text = "" then
Msgbox "digite nombre para borrar rregistro", vbcritical, "hipermarket"
Else
On error resume next
If err.number = 3021 then
Msgbox "no hay ningun registro para borrar", vbinformation, "hipermarket"
Else
Dataclave.refresh
Dataclave.recordset.movelast
Do while not dataclave.recordset.eof
On error resume next
If err.number = 94 then
Dataclave.recordset.fields("nombre usuario") = ""
Else
Exit do
Dataclave.recordset.movelast
End if
Loop
Dataclave.recordset.delete
Msgbox "su registro se ha borrado satisfactoriamente", vbinformation, "hipermarket"
End if
End if
Txclave.text = ""
Dbcusuario.text = ""
End sub
```

---

```
Private sub dbcusuario_click(area as integer) 3
```

```
If dbcusuario.text = "" then
Dbcusuario.setfocus
Else
Txclave.setfocus
End if
End sub
```

---

```
Private sub dbcusuario_keypress(keyascii as integer) 3
If keyascii = 13 then
Txclave.setfocus
End if
End sub
```

---

```
Private sub okbutton_click() 5
Beep
Bandera = true
If dbcusuario.text = "" or txclave.text = "" then
Msgbox "no deje campos vacios", vbcritical, "hipermarket"
Else
Dataclave.refresh
Dataclave.recordset.movefirst
Do while not dataclave.recordset.eof
If dataclave.recordset.fields("nombreusuario") = dbcusuario.text and
dataclave.recordset.fields("contraseña") = txclave.text then
Bandera = true
Frmsplash.show
Exit do
Else
Dataclave.recordset.movenext
Bandera = false
End if
Loop
End if
If bandera = false then
Msgbox "clave incorrecta, por favor inténtelo de nuevo", vbinformation, "atención"
Txclave.text = ""
Txclave.setfocus
C = c + 1
End if
If c = 3 then
Msgbox "imposible ingresar a falcon administrador", vbcritical, "error"
End
End if
End sub
```

---

```
Private sub txclave_keypress(keyascii as integer) 6
If keyascii = 13 then
Okbutton.value = true
End if
End sub
```

Programacion imagen dos

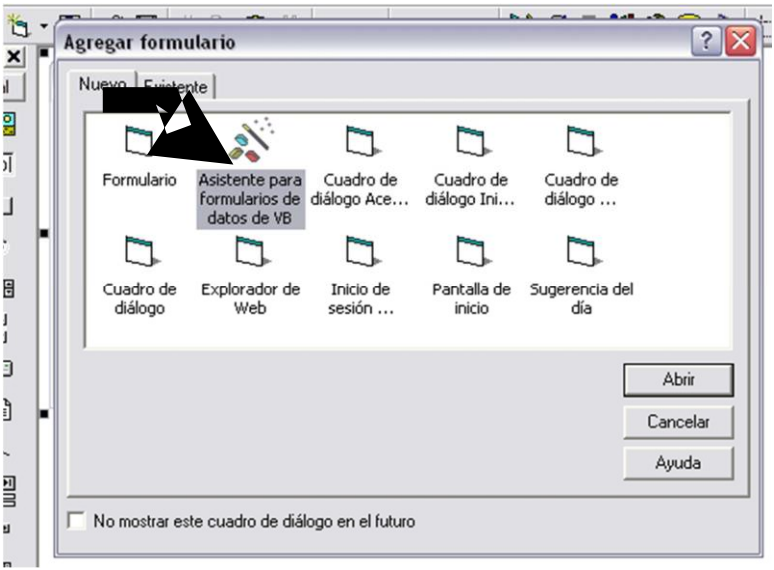
```
Private sub form_load()  2
Unload frmlogin
Move (screen.width - width) / 2, (screen.height - height) / 2
End sub

Private sub timer1_timer() 1
Mdiform1.show
End sub
```

Programación de la imagen tres

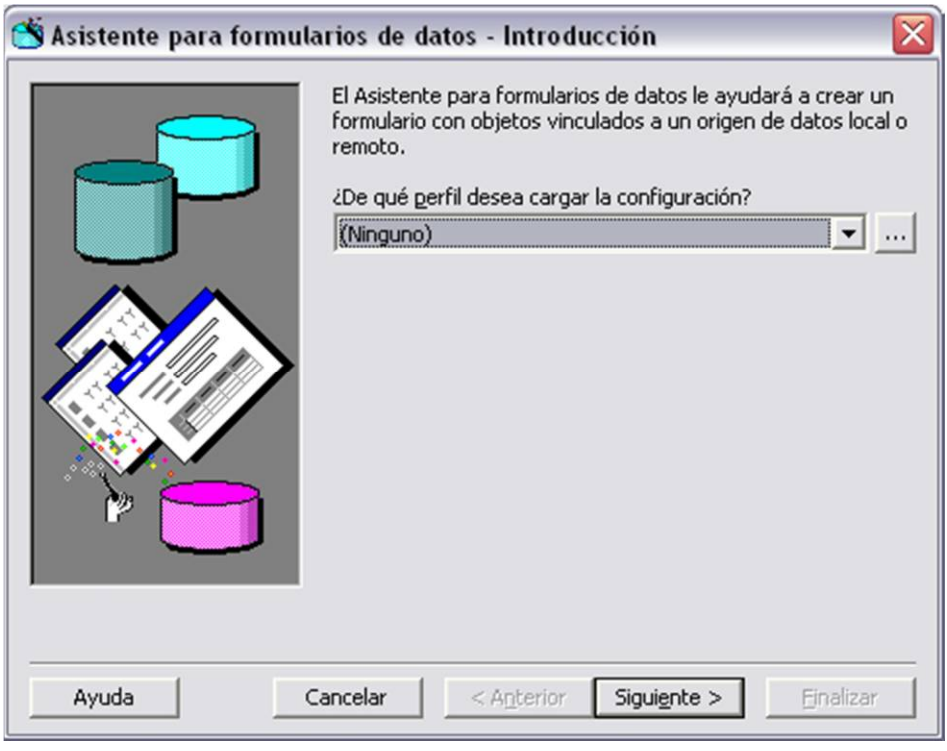
```
Private sub mdiform_load()
Unload frmsplash
End sub
```

# 31. CREACION DE AUTO ASISTENTES

Se utilizan en caso que sean tablas de solo llenado para que visual programe automáticamente solo bastaría con cambiarle el nombre a las cajas de texto y crearles los focos para hacerlo es de la siguiente manera

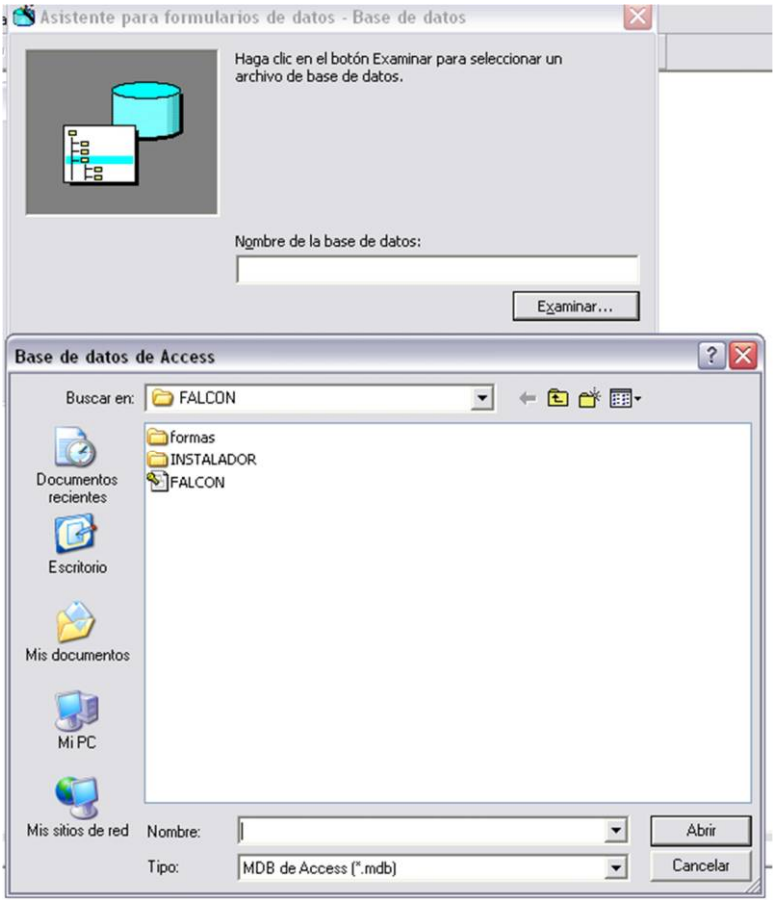



Click donde indica la flecha luego asistente para formularios de datos vb esto significa visual Basic luego click en abrir

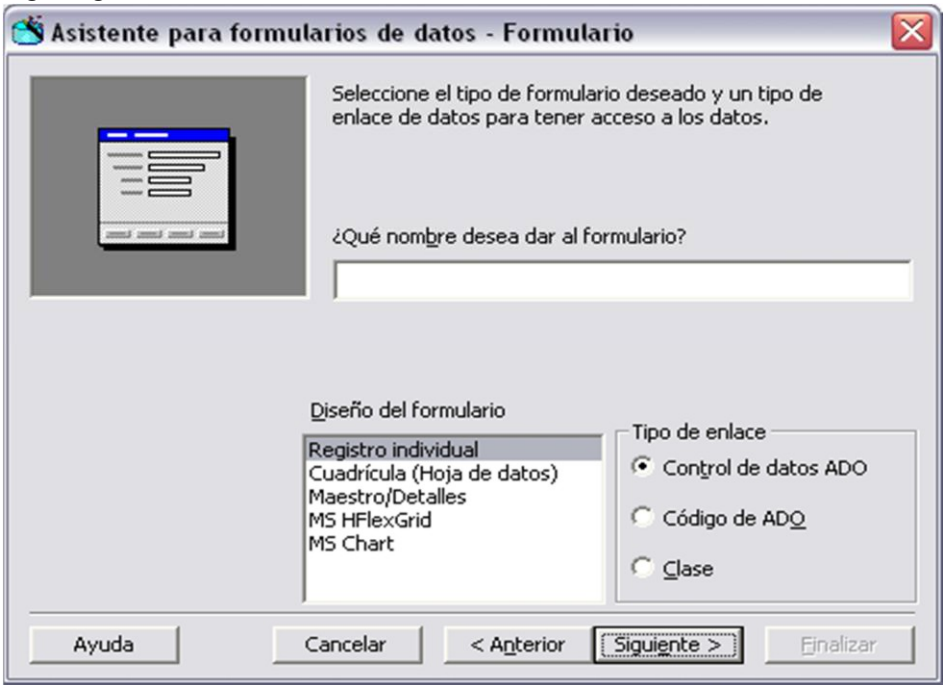



Luego click en siguiente

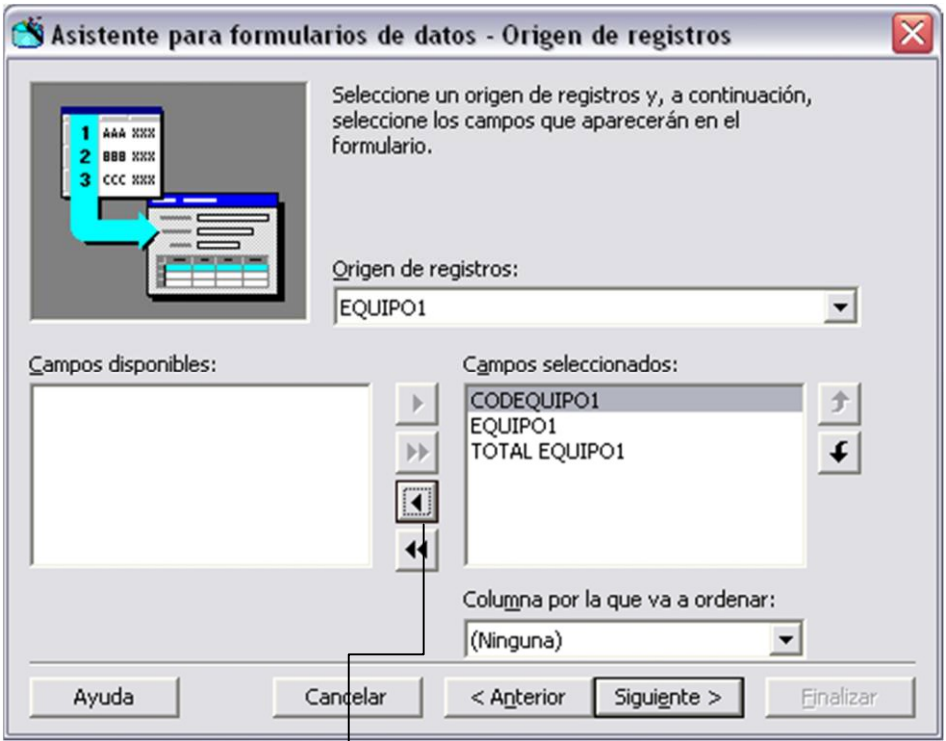
Elijo Access luego siguiente




Luego click en el botón examinar y elijo la ruta donde está la base de datos y abrir luego siguiente



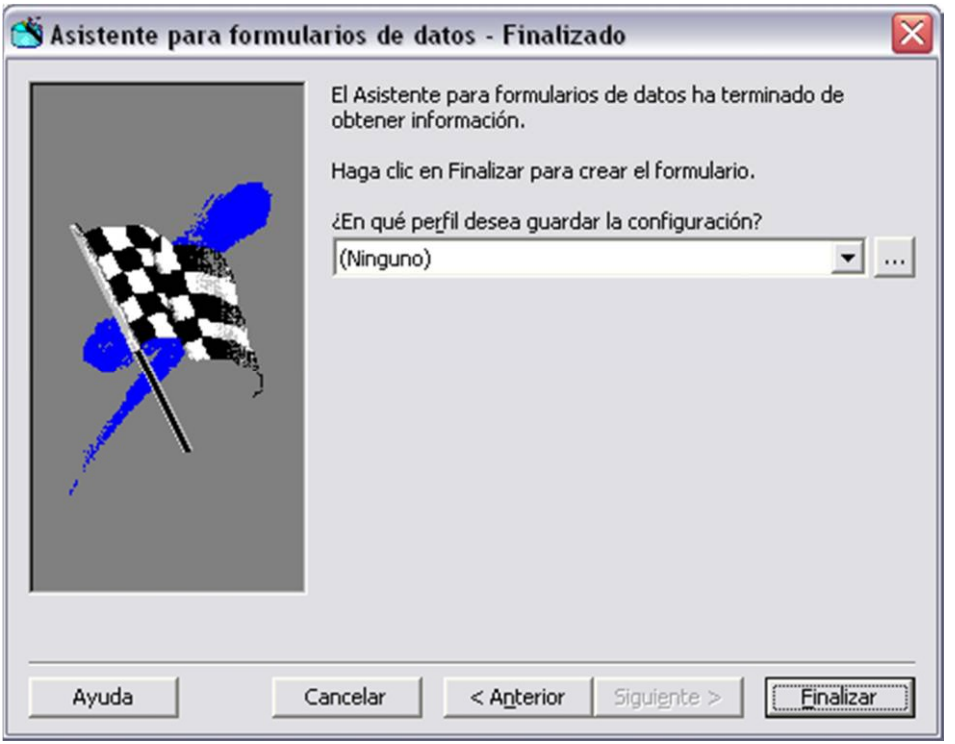
Luego sin tocar nada en el cuadro de dialogo anterior siguiente




En origen de registros elijo la tabla a la cual le voy hacer autoformulario luego agrego todos los campos y siguiente

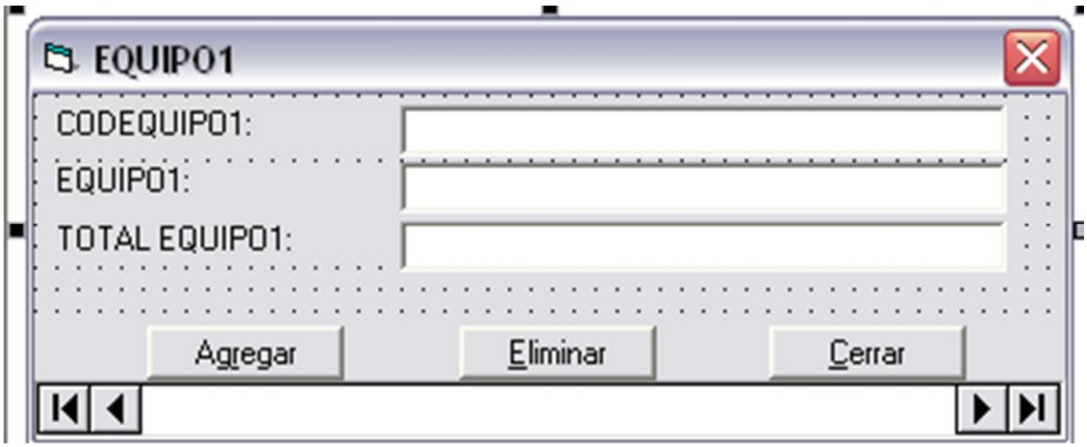
Luego quito la verificación actualizar y de renovar y siguiente

Luego finalizar



Luego aceptar

Y quedara así




Este ya está todo programado solo bastaría crear los focos de cada caja de texto un ejemplo de asistente ya terminado se vería así

El foco seria en el evento keypress de cada caja hago al parecido

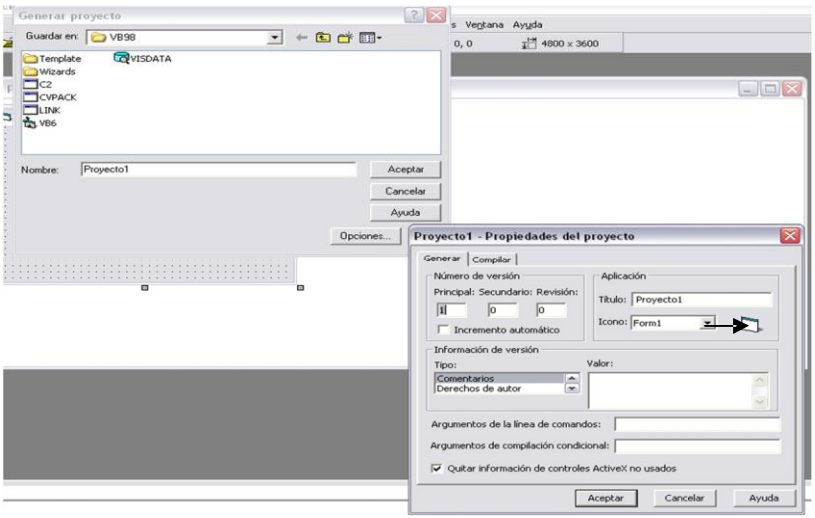
Private sub text1_keypress(keyascii as integer)
If keyascii = 13 then
Text1(1).setfocus  ese uno demás significa que es la misma caja pero con control de matrices es el índice
End if
End sub

El **(1)** es un índice del **arrayz o vector** que utilizan las cajas de texto de estos autoasistentes

# 32. CREACION DEL EJECUTABLE

Después de que un proyecto está configurado toca crear el punto exe de la siguiente manera

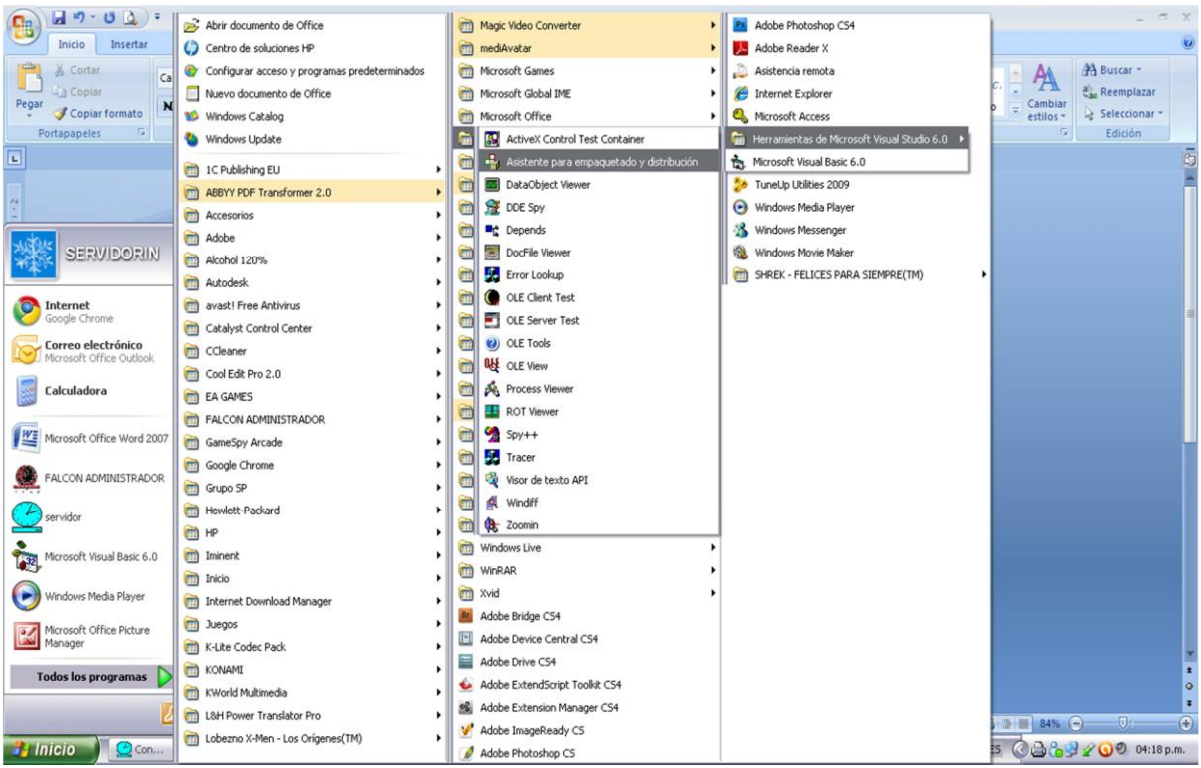
Generar proyecto.exe



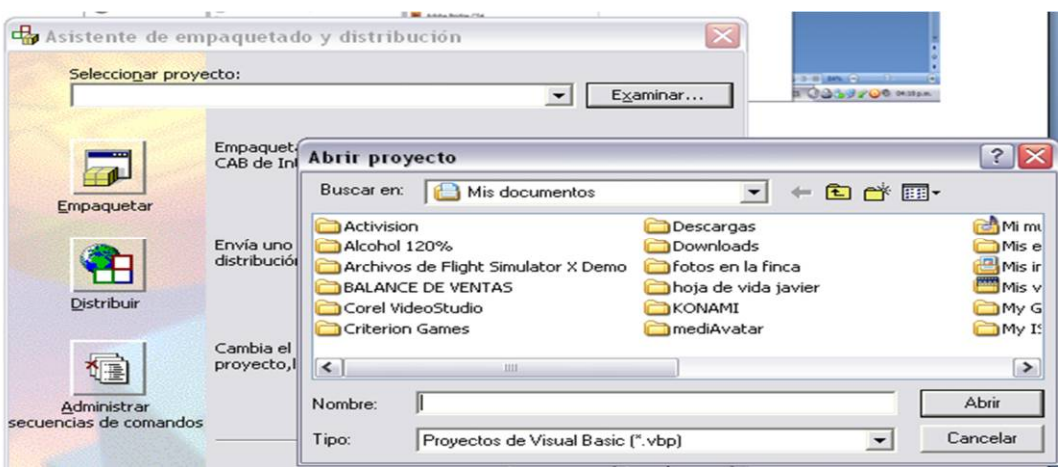
Luego opciones y cambiar el icono que indica la flecha este icono es de acuerdo al que le haya cargado a cualquier formulario en la parte superior y aceptar hemos creado el ejecutable del programa

## 33. INSTALADOR

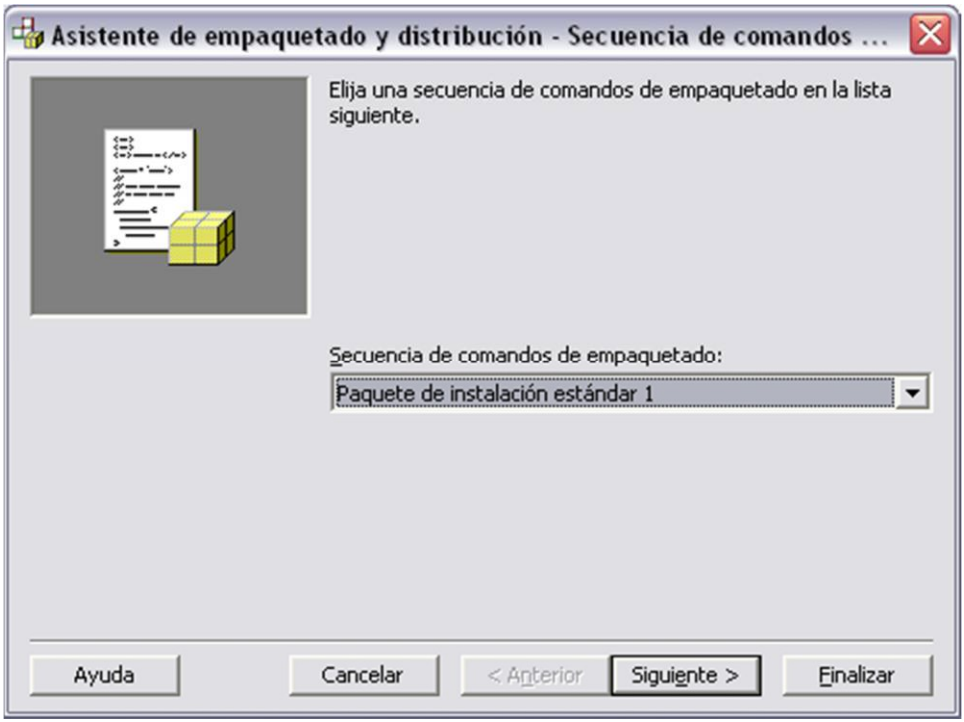
Ahora haremos el instalador del programa




Inicio todos los programas herramientas de Microsoft visual studio 6.0 y luego asistente para empaquetado y distribución

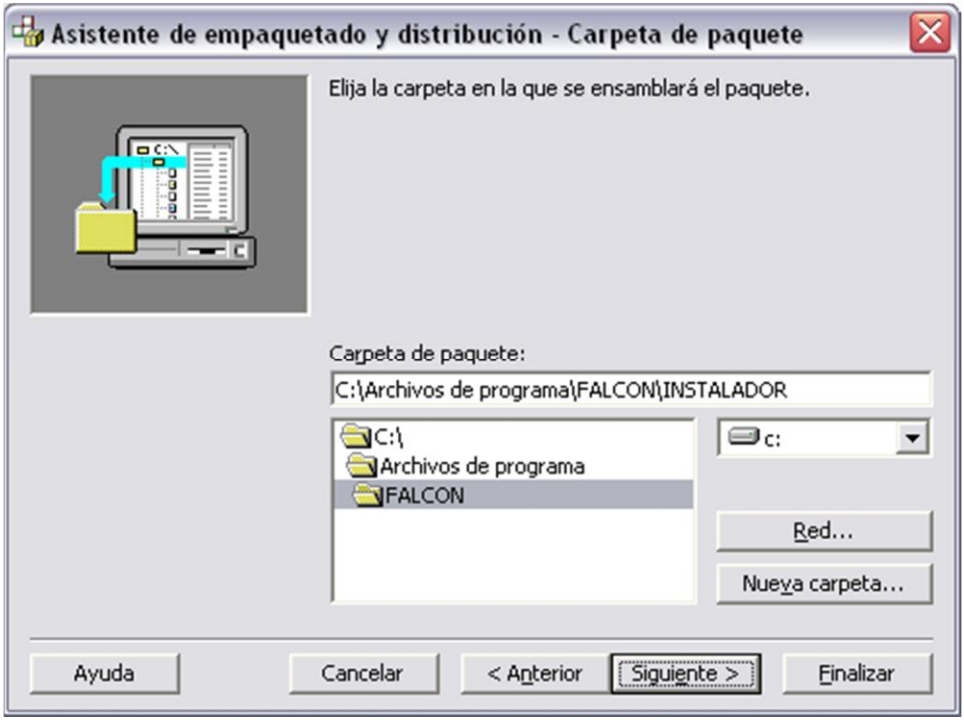


Luego click en examinar y voy y busco el.exe del programa que acabe de crear anteriormente todo esto se guarda en la carpeta del programa donde se programó luego click en el botón empaquetar




Luego siguiente

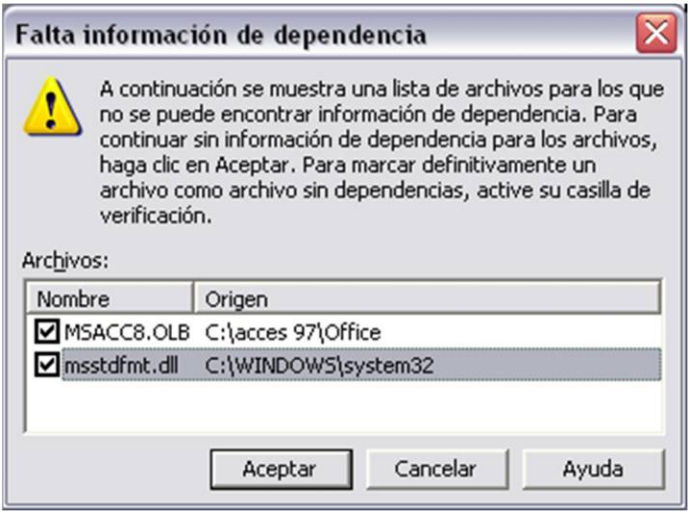
Luego elegir paquete de instalación estándar y siguiente




Creo una carpeta nueva en el botón nueva carpeta la selecciono y siguiente

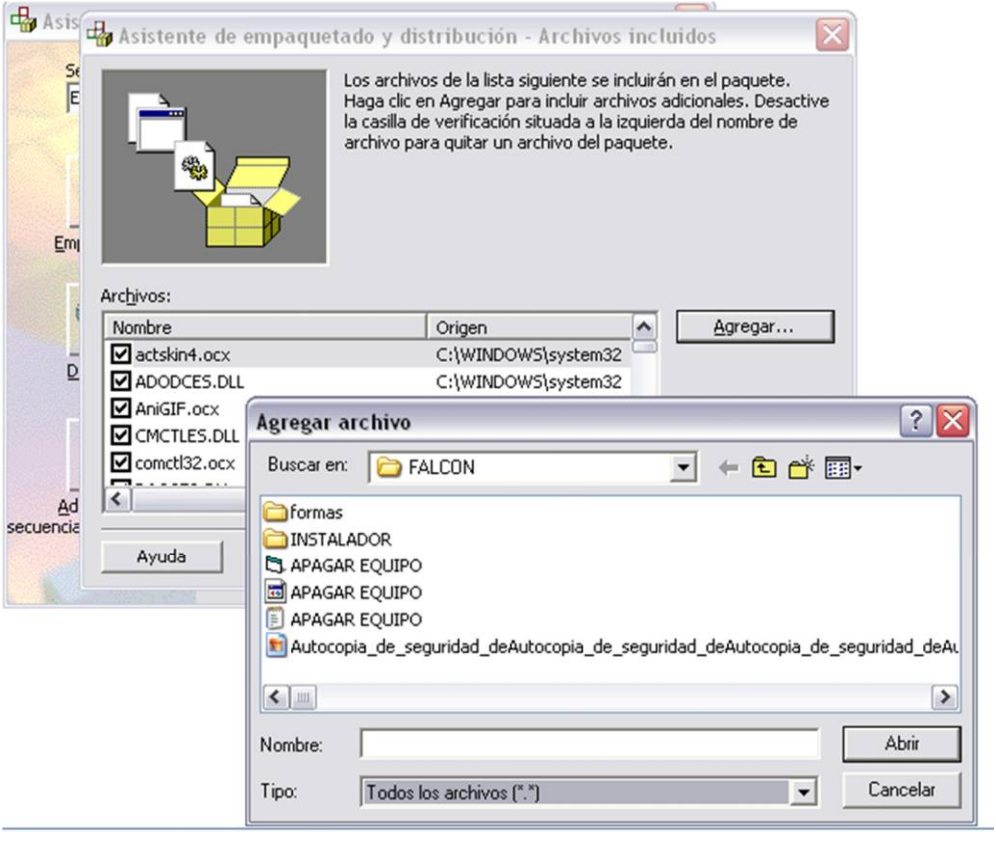
Luego agrego lo siguiente

Luego siguiente




Es posible que este cuadro no salga

Luego chuleo esto y aceptar

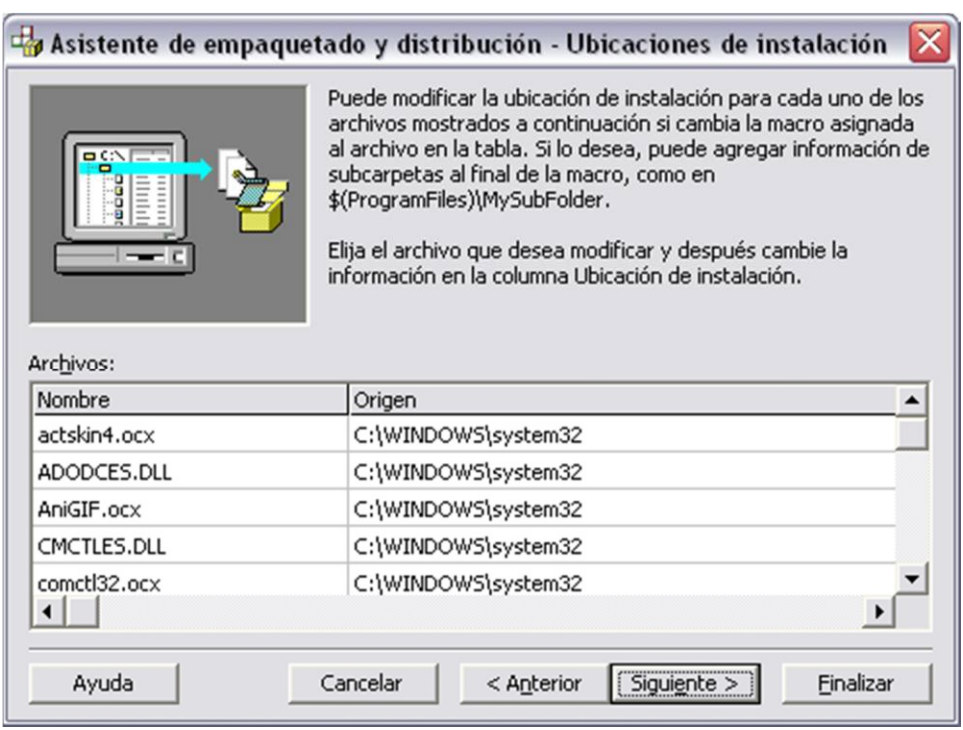


Luego click en agregar y agrego los controles, que **utlize la base de datos** y **el punto exe** del programa seleccionándolo y abriéndolo luego siguiente

Luego siguiente

Le damos el nombre al instalador luego siguiente

Luego siguiente




Luego siguiente

Luego chuleo todo lo que me muestra aquí y siguiente

Luego finalizar y listo

# 34. MANEJO DE INTERNET, WORD, EXCEL DESDE VISUAL

A continuación como cargar un archivo de Word, Excel y como cargar internet desde un botón

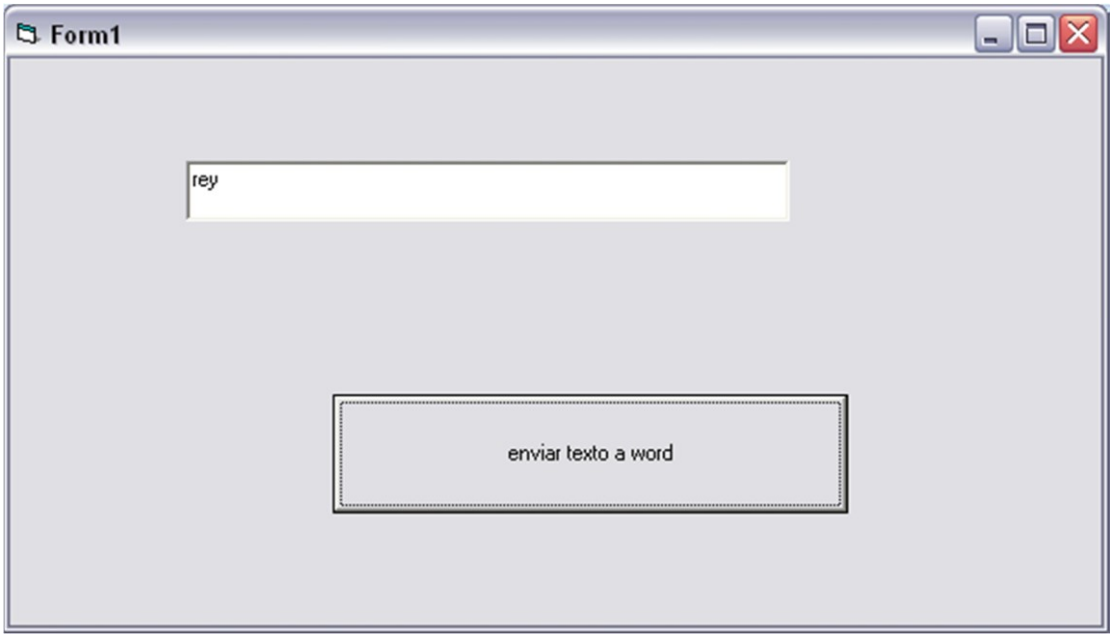
## Para internet

```
Dim intobj as object
On error resume next
If err.number = 450 then
Else
Set intobj = createobject("internetexplorer.application")
Intobj.visible = -1
Intobj.navigate
Do until intobj.busy = false
Loop
End if
```

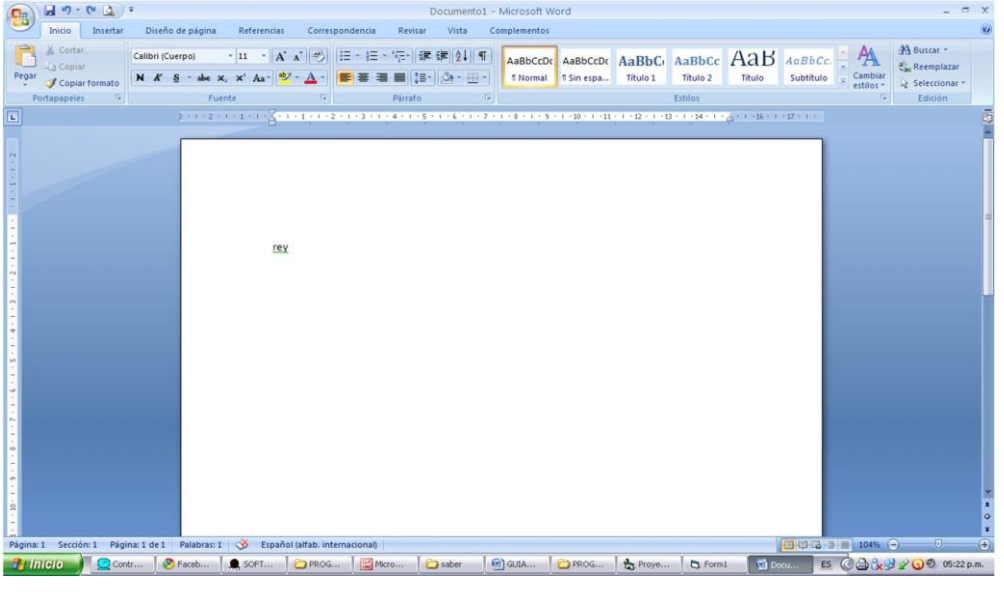
## Para word

```
On error goto err_comando8_click

Dim oapp as object

Set oapp = createobject("word.application")
Oapp.visible = true

Exit_comando8_click:
Exit sub

Err_comando8_click:
Msgboxerr.description
Resume exit_comando8_click
```

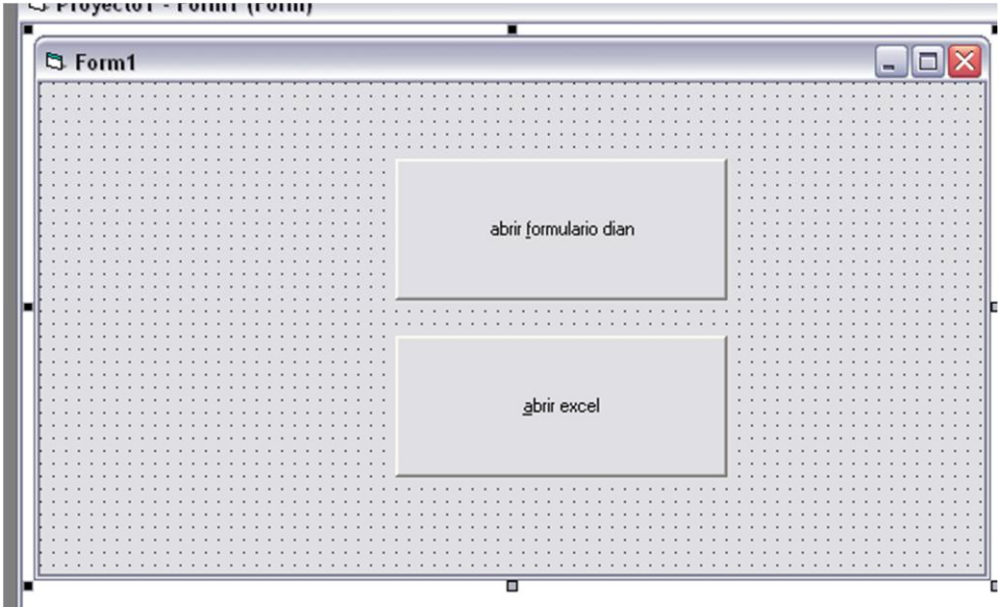
## Para enviar lo que está en una caja de texto a Word




```
Dim msword as new word.application
Dim document as word.document
Dim parrafo as paragraph

Set documento = msword.documents.add
Set parrafo = documento.paragraphs.add

Parrafo.range.insertafter text1.text

Msword.visible = true
```



## Para abrir un archivo de  Excel




Esto es para abrir un archivo existente de Excel
Private sub command1_click()
Dim objexcel as excel.application
Dim xlibro as excel.workbook
Set objexcel = new excel.application
Set xlibro = objexcel.workbooks.open("c:\archivos de programa\saber\formulario retefuente.xlsx")
Objexcel.visible = true
End sub

---

## Y esto es para abrir Excel

Private sub command2_click()
Dim objexcel as excel.application
Dim xlibro as excel.workbook
Set objexcel = new excel.application
Set xlibro = objexcel.workbooks.open("c:\archivos de programa\saber\libro1.xlsx")
Objexcel.visible = true
End sub

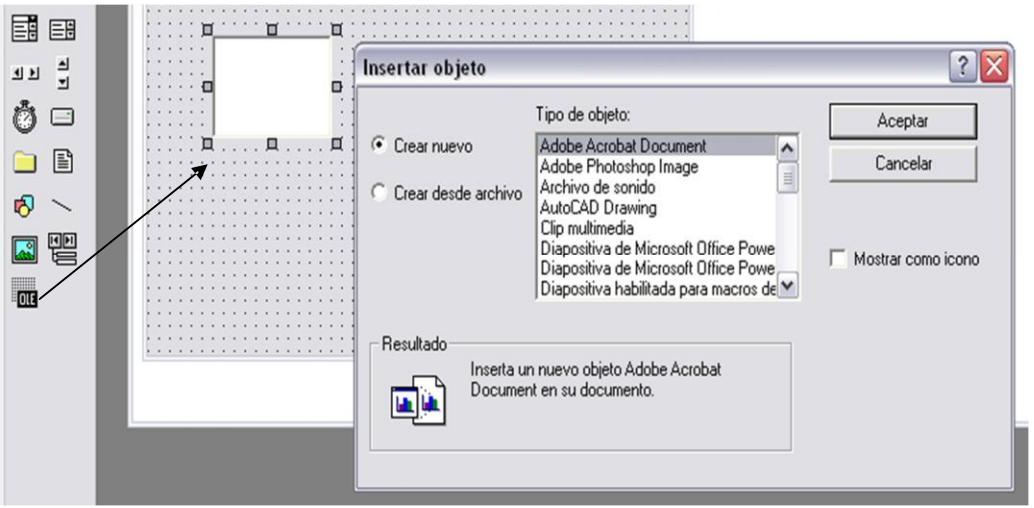
Para lo de Excel, Word hay que cargar unas referencias de la siguiente manera

Menú proyecto referencias, cuando es office 2007 es **12.0 objectlibrary** cuando es office 2010 **14.0 objectlibrary** cuando es office 2003 es **11.0 objectlibrary** cuando es office 97 es **8.0 objectlibrary**

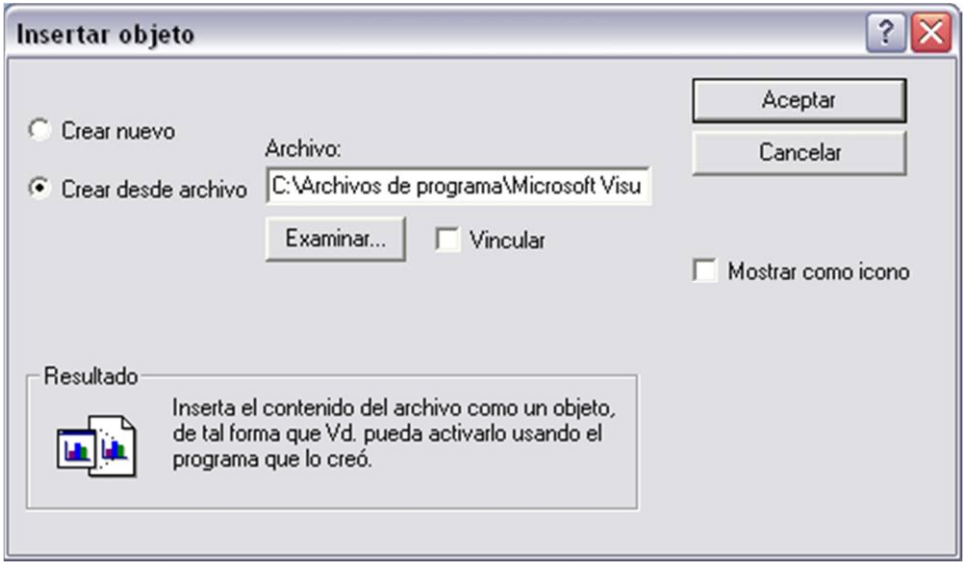
El ejemplo es practicar y averiguar cómo se carga PowerPoint

## 35. OBJETO OLE

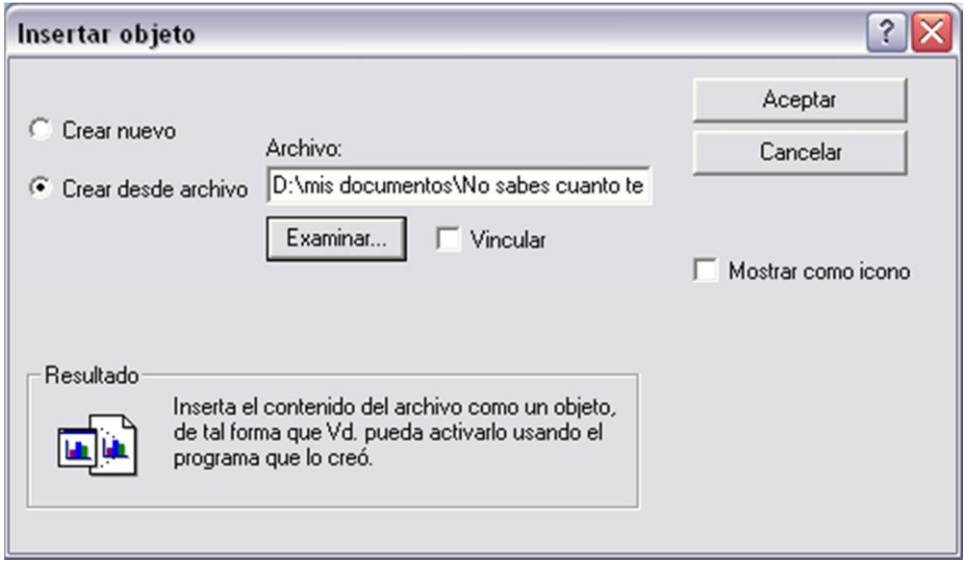
Utilización de un objeto ole para incrustar un archivo de office

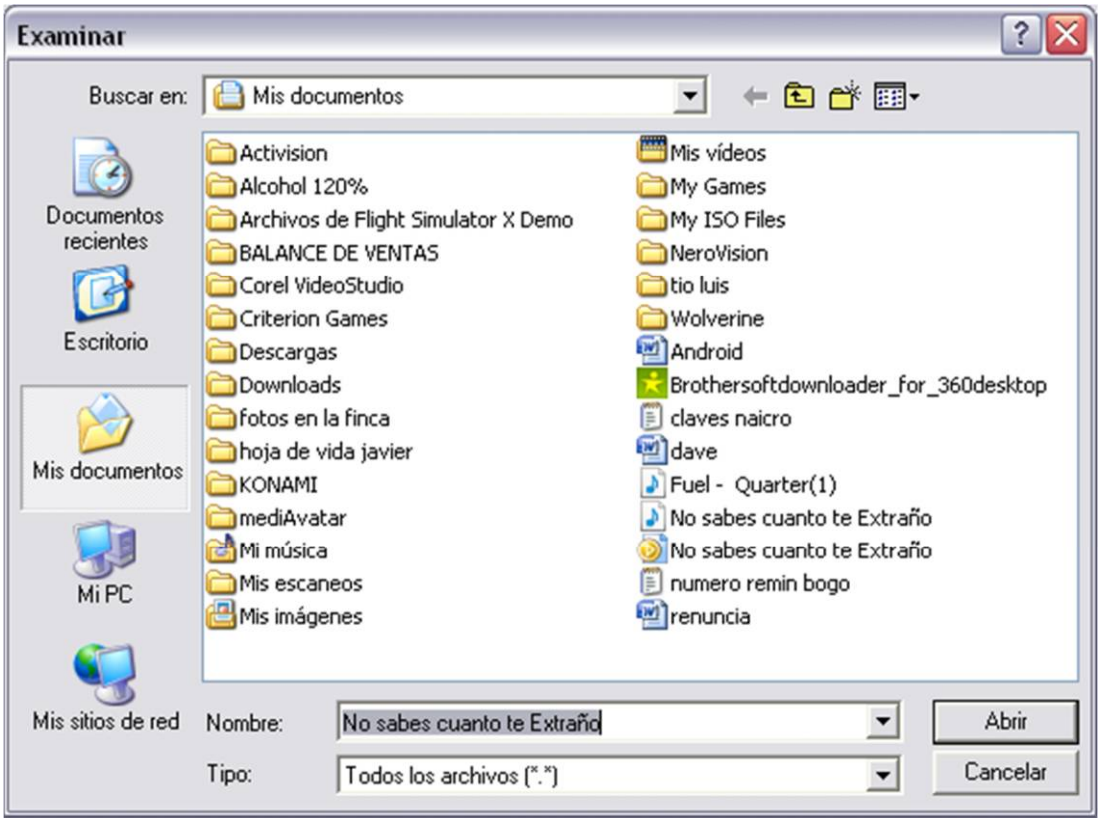



Elijo el tipo de aplicación que voy a insertar o si no lo insertamos desde archivo




Si es desde archivo doy click en el botón examinar





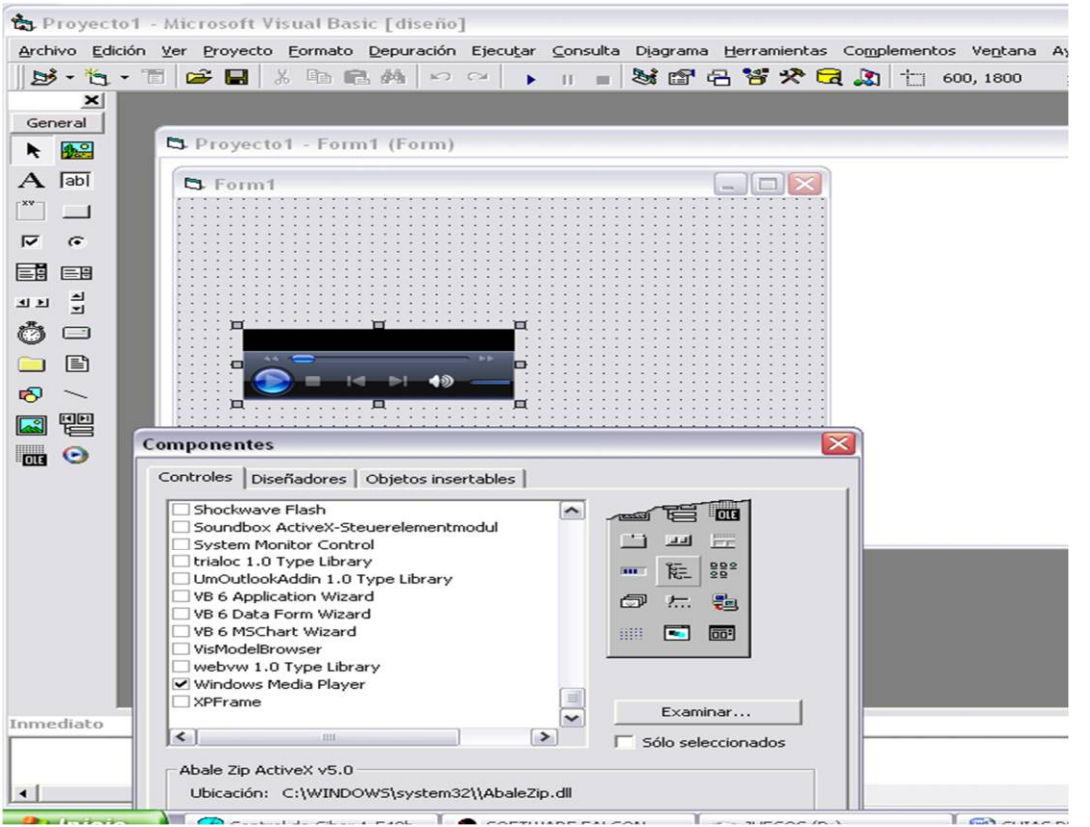

Elijo el archivo y abrir

Luego aceptar

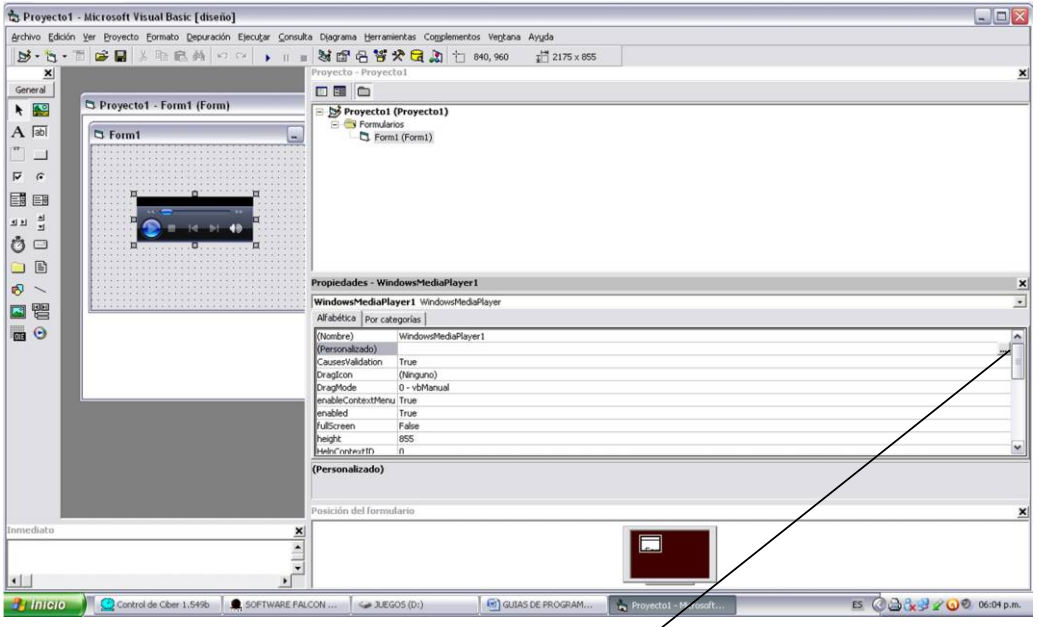
En tiempo de ejecución se dará doble click sobre el objeto ole

# 36. VIDEOS A VISUAL BASIC

## Para insertar un video visual Basic




Click derecho sobre la barra de controles luego componentes luego chuleamos la opción **Windows media player** y aceptar para programar el control haremos esto

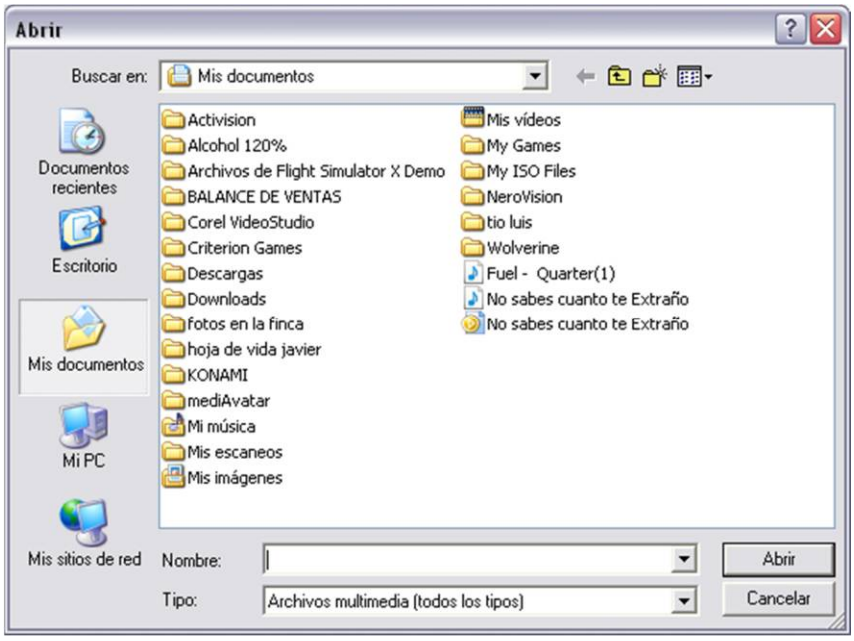


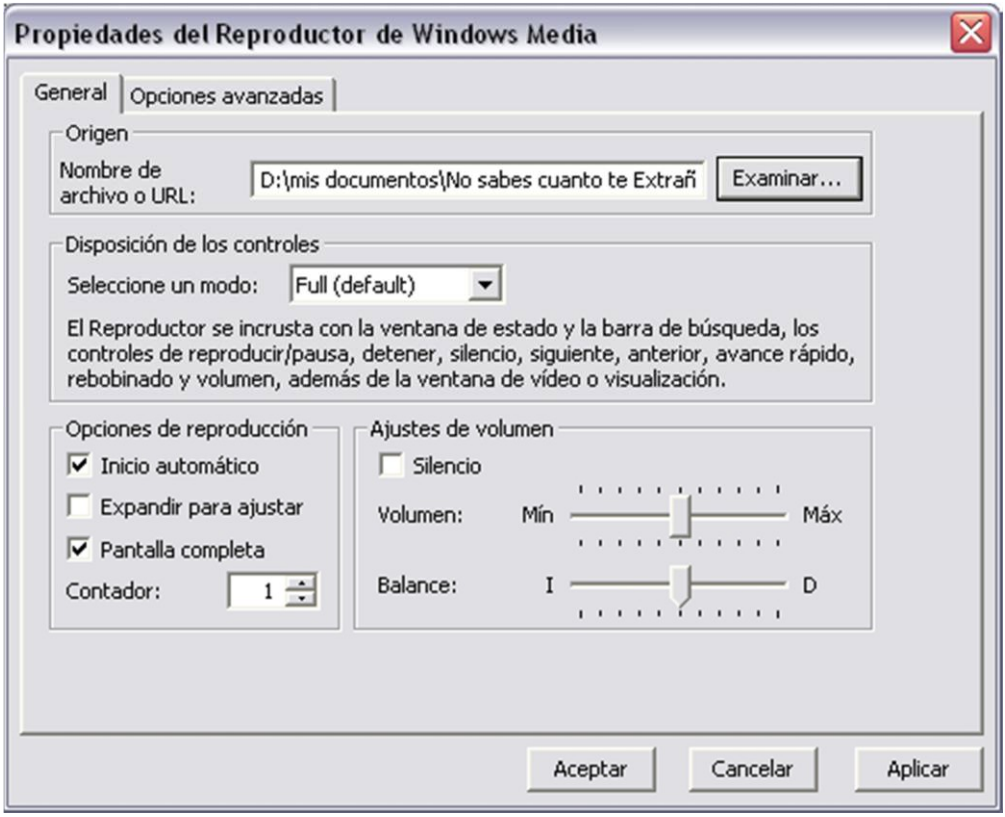

Click en la propiedad personalizado

Luego sale esto

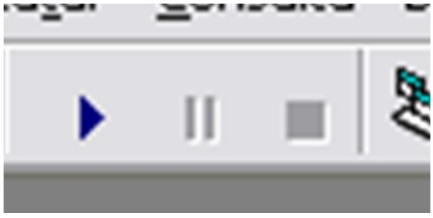
Donde dice url doy click en el botón examinar para cargar el video

Luego selecciono el video y aceptar luego

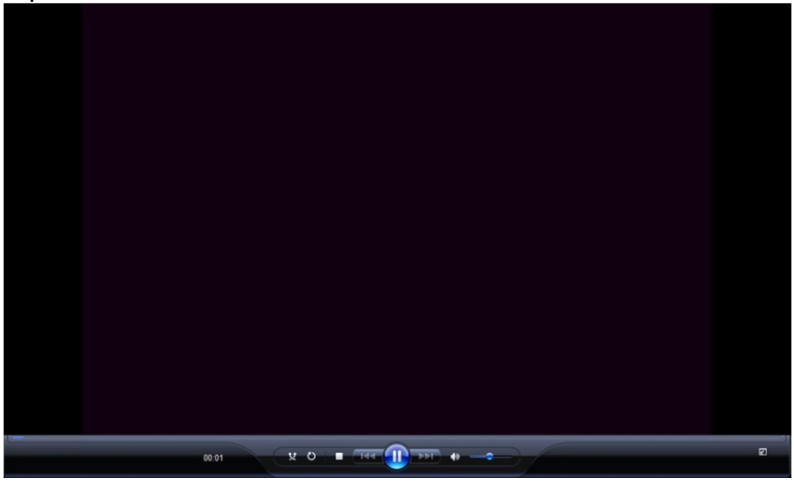





Chuleo las casillas inicio automático y pantalla completa luego aplicar y aceptar luego para ver el video ejecuto el proyecto ya sea con f5 o la barra play

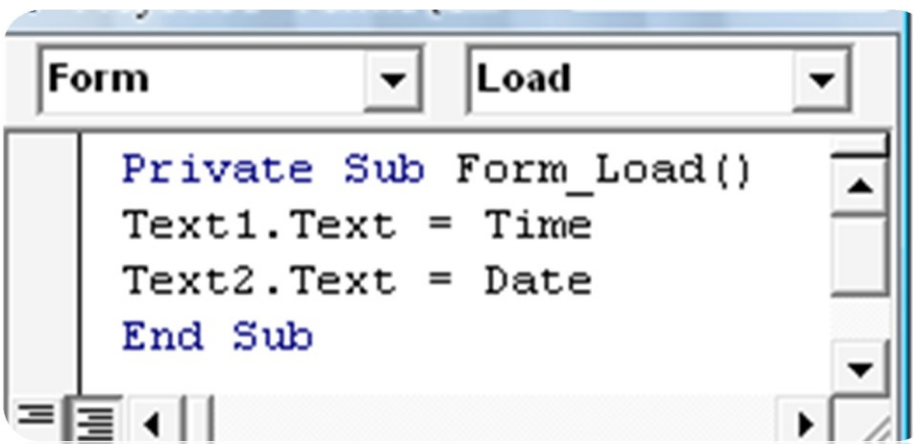


Y quedara así

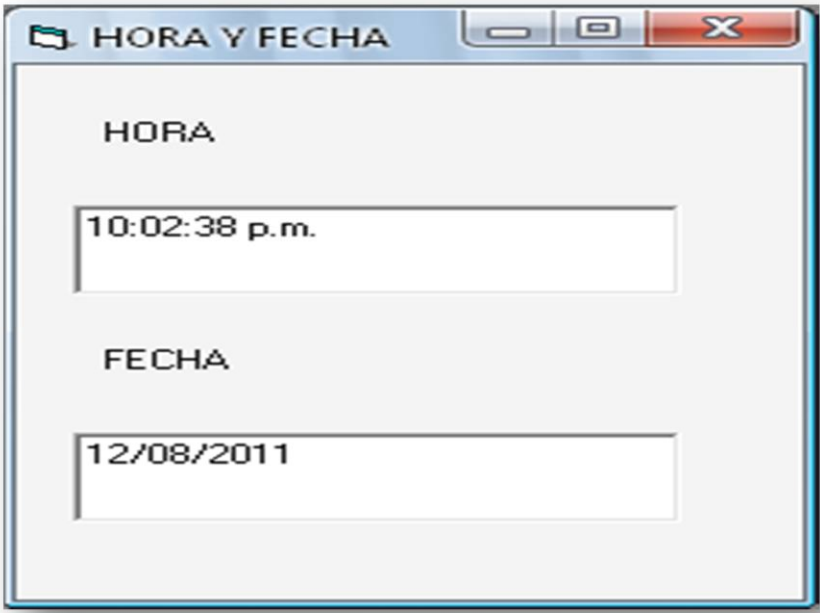


# 37. CARGAR LA HORA Y LA FECHA A UNA CAJA DE TEXTO

La programación de la siguiente aplicación es:



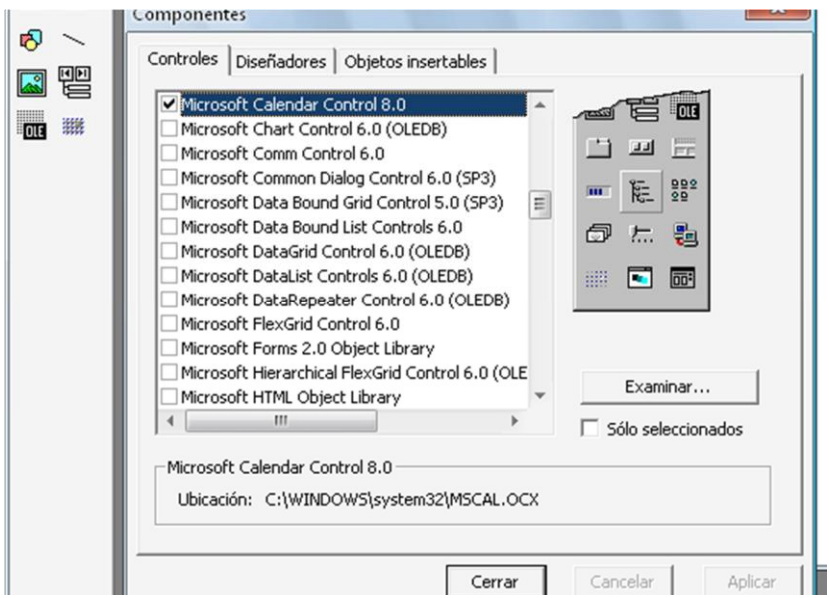
En ejecución queda así:



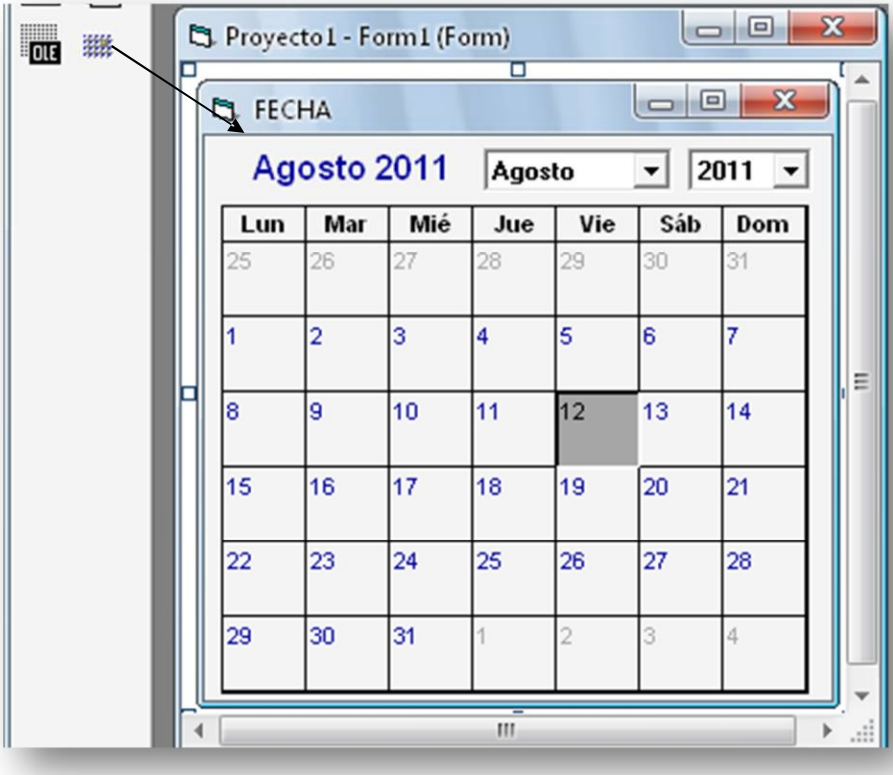

## 38. CARGAR CONTROL CALENDARIO

Click derecho sobre la barra de controles referencias y sacar el siguiente control

**MICROSOFT CALENDAR CONTROL 8.0** tal como se muestra en la gráfica



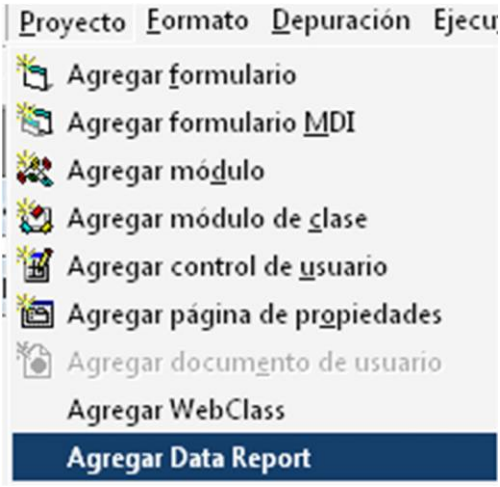
Luego dibujo el control sobre el formulario así:



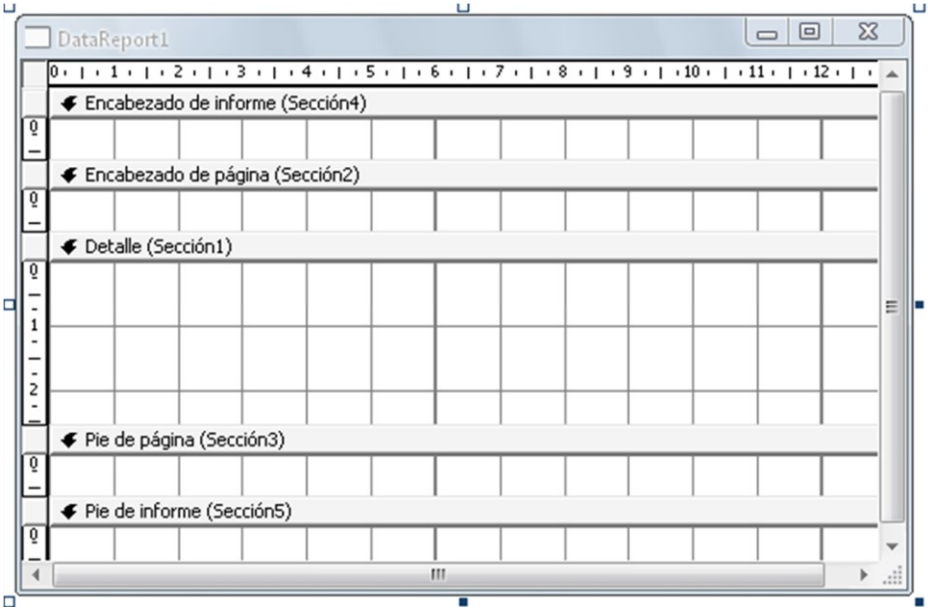

En ejecución quedara así:

# 39. INFORMES DESDE VISUAL UTILIZANDO DATA ENVIRONMENT Y DATA REPORT

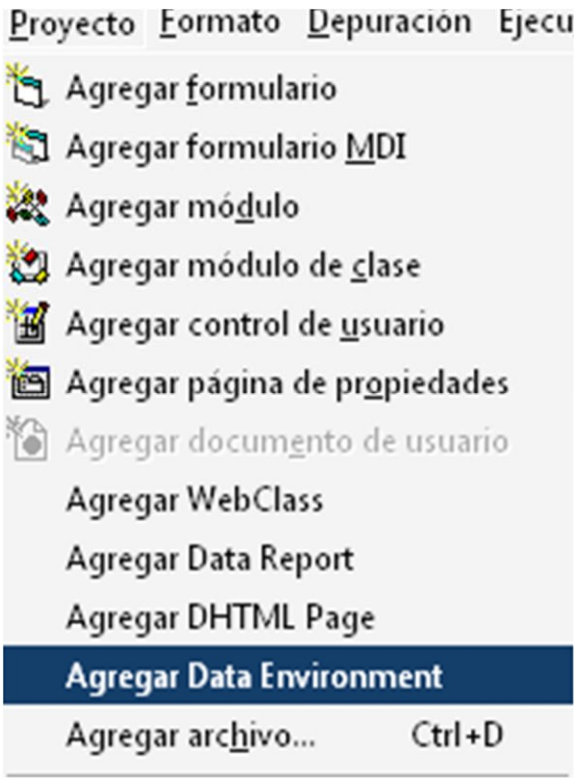
Click en el menú proyecto luego agregar data report



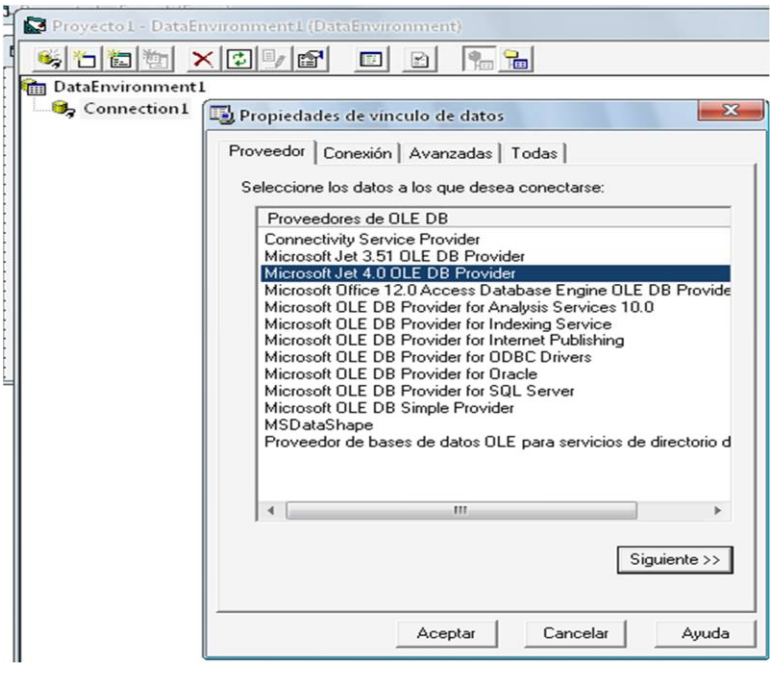

Luego aparece el siguiente formulario



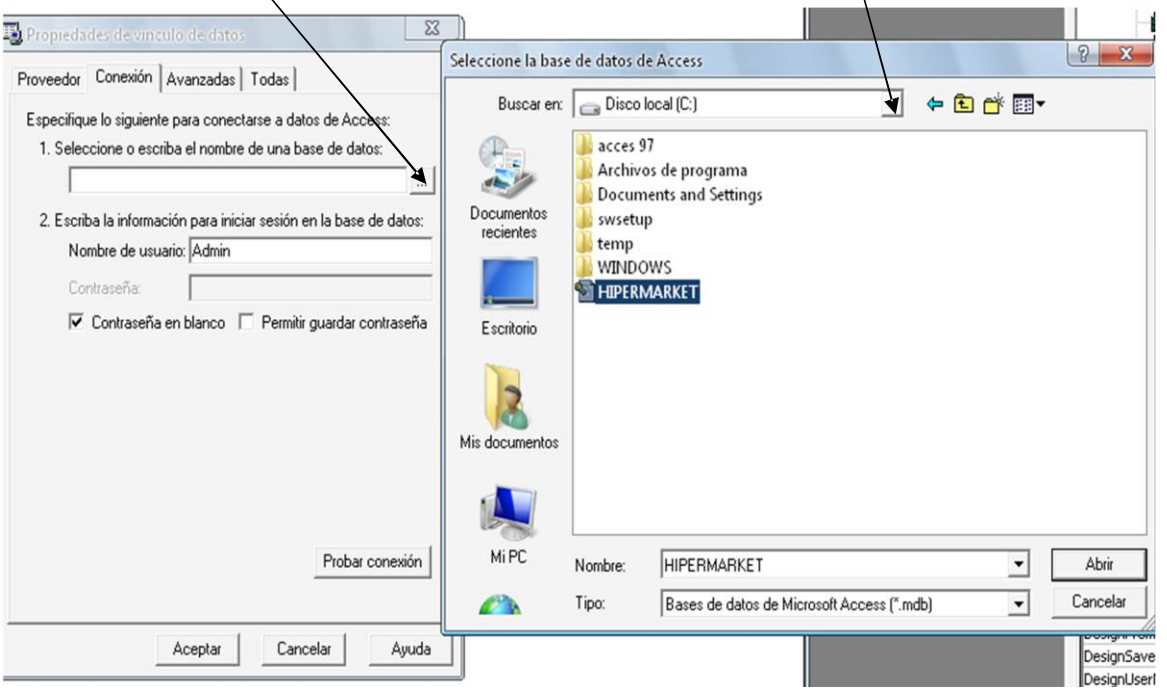
Luego click en el menú proyecto agregar data environment



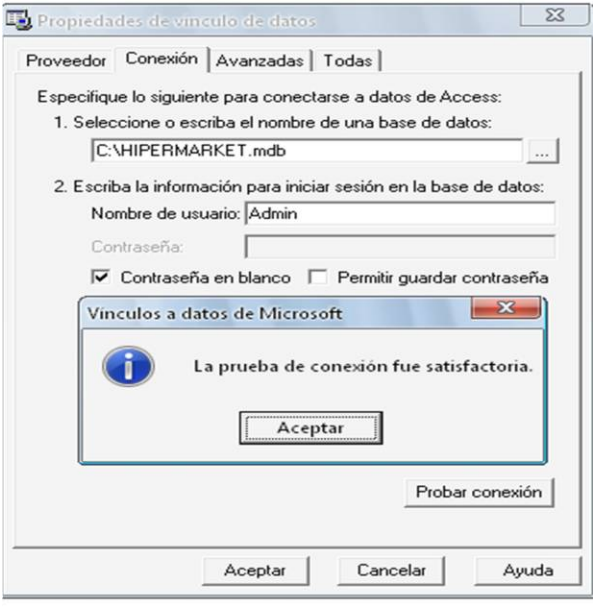

Luego click derecho en connection1 y elegir la opción **MICROSOFT JET 4.0 OLE DB PROVIDER** Luego siguiente



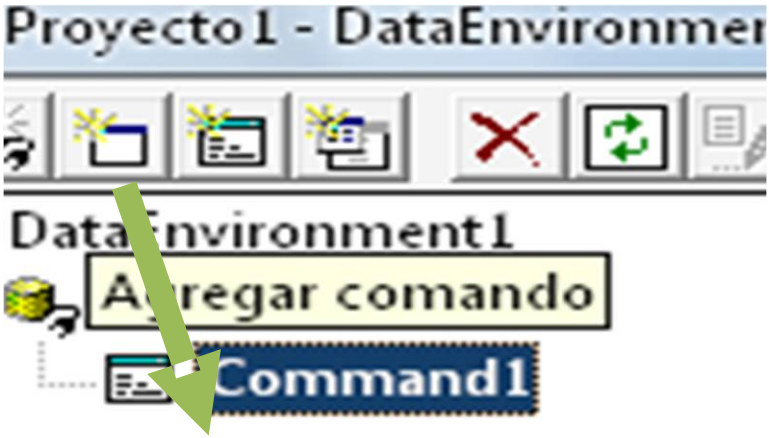
Luego click aquí para cargar la base de datos elijo la ruta aquí y luego abrir



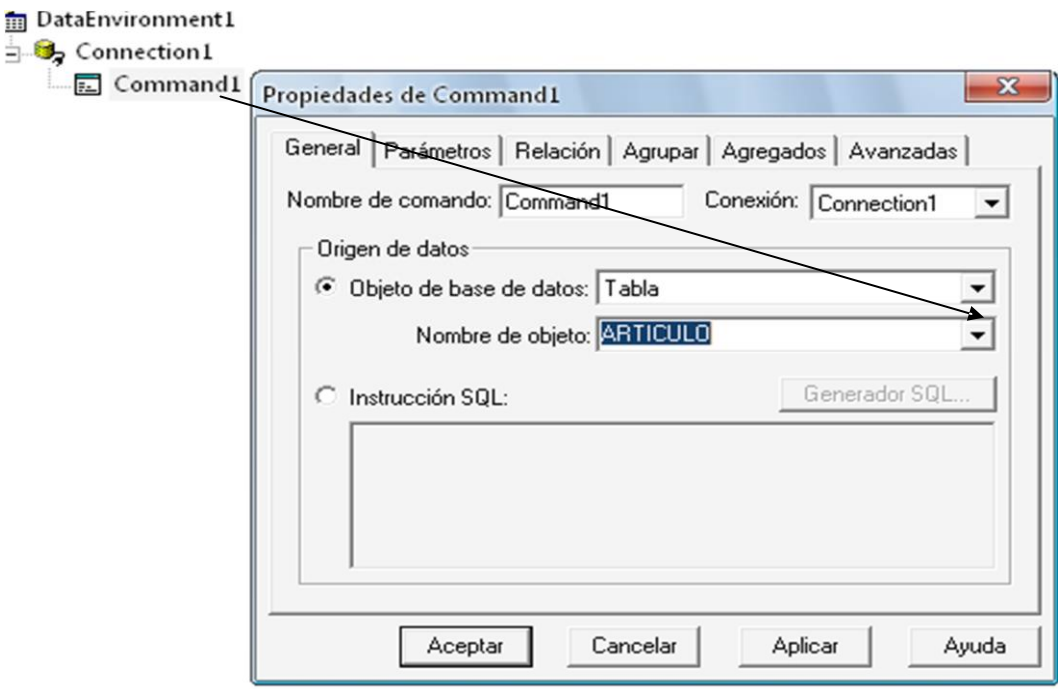

Luego click en el botón probar conexión Y luego aceptar



Luego click en agregar comando tal como indica la gráfica

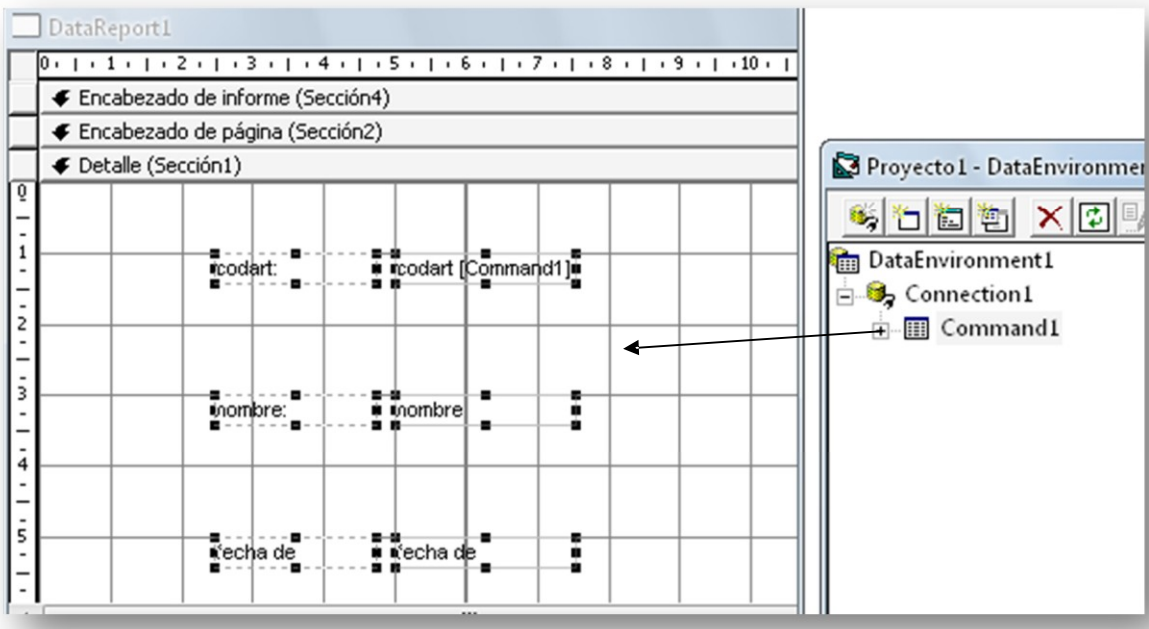



Luego click derecho sobre command1, luego en objeto de base de datos elijo tabla luego en nombre del objeto escojo el nombre de la entidad a elaborar el informe luego aplicar y aceptar

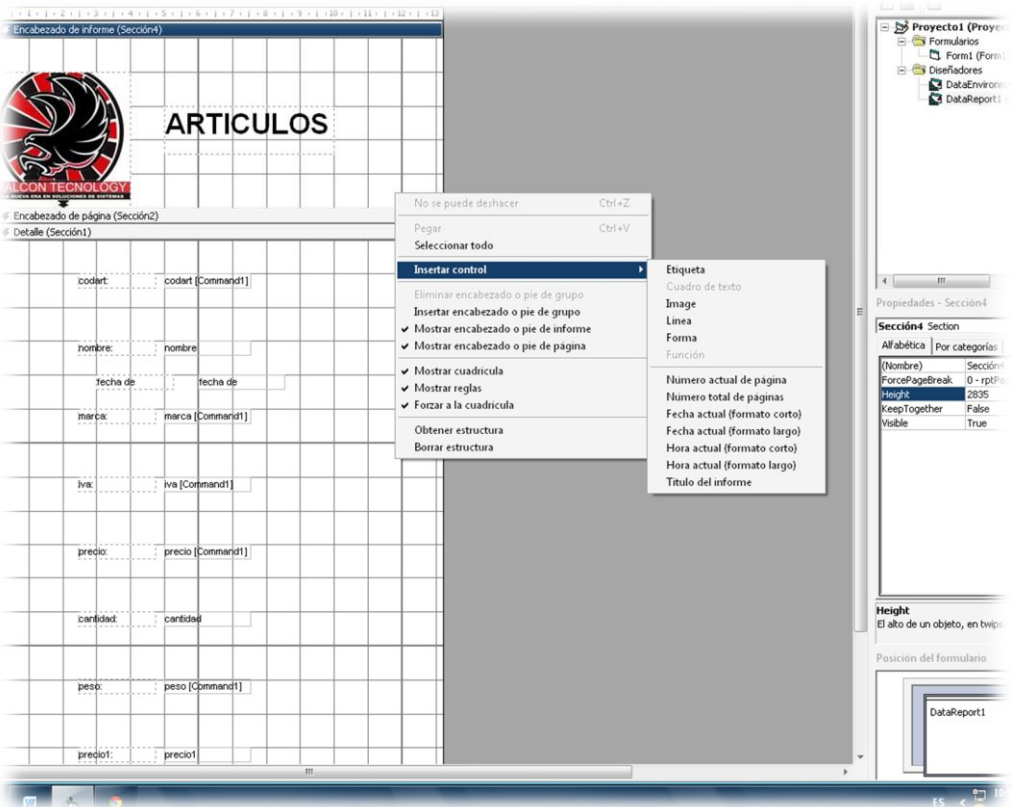


Luego quedan los campos tal y como muestra la gráfica

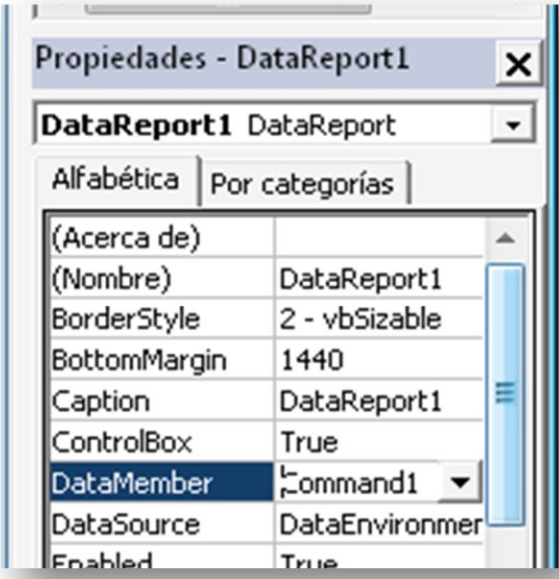
Luego arrastro el command1 al data report tal como muestra la gráfica



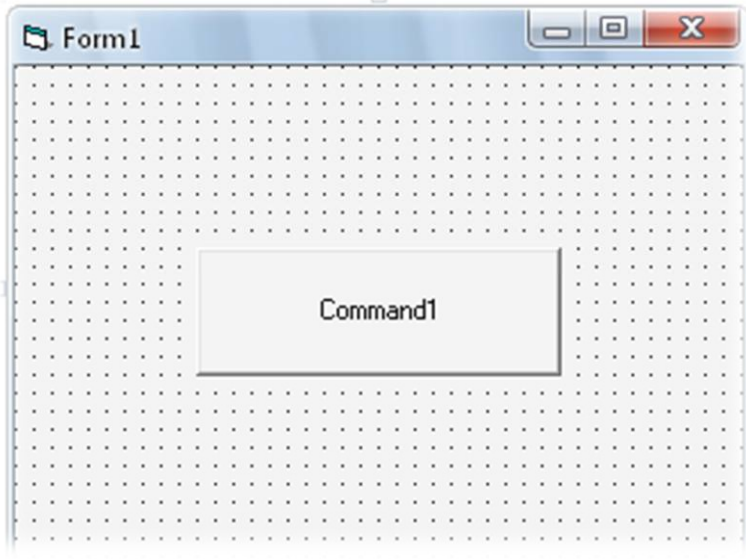
Luego ordeno los campos en el data report para que quede bonito, en la propiedades del data report modifico lo siguiente




click derecho para insertar los controles para cargar las  imágenes las etiquetas etc…



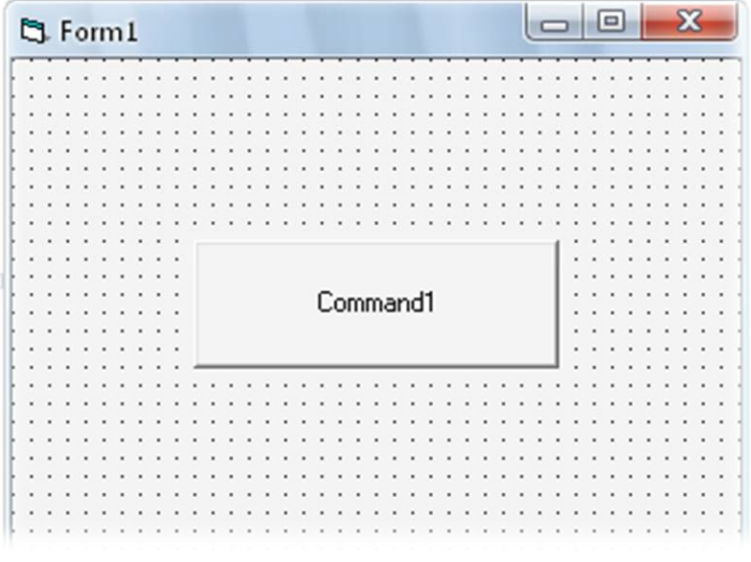

En **DATAMEMBER** cargo el command1, y en **DATASOURCE** cargo el dataEnvironment luego coloco otro formulario con un command button



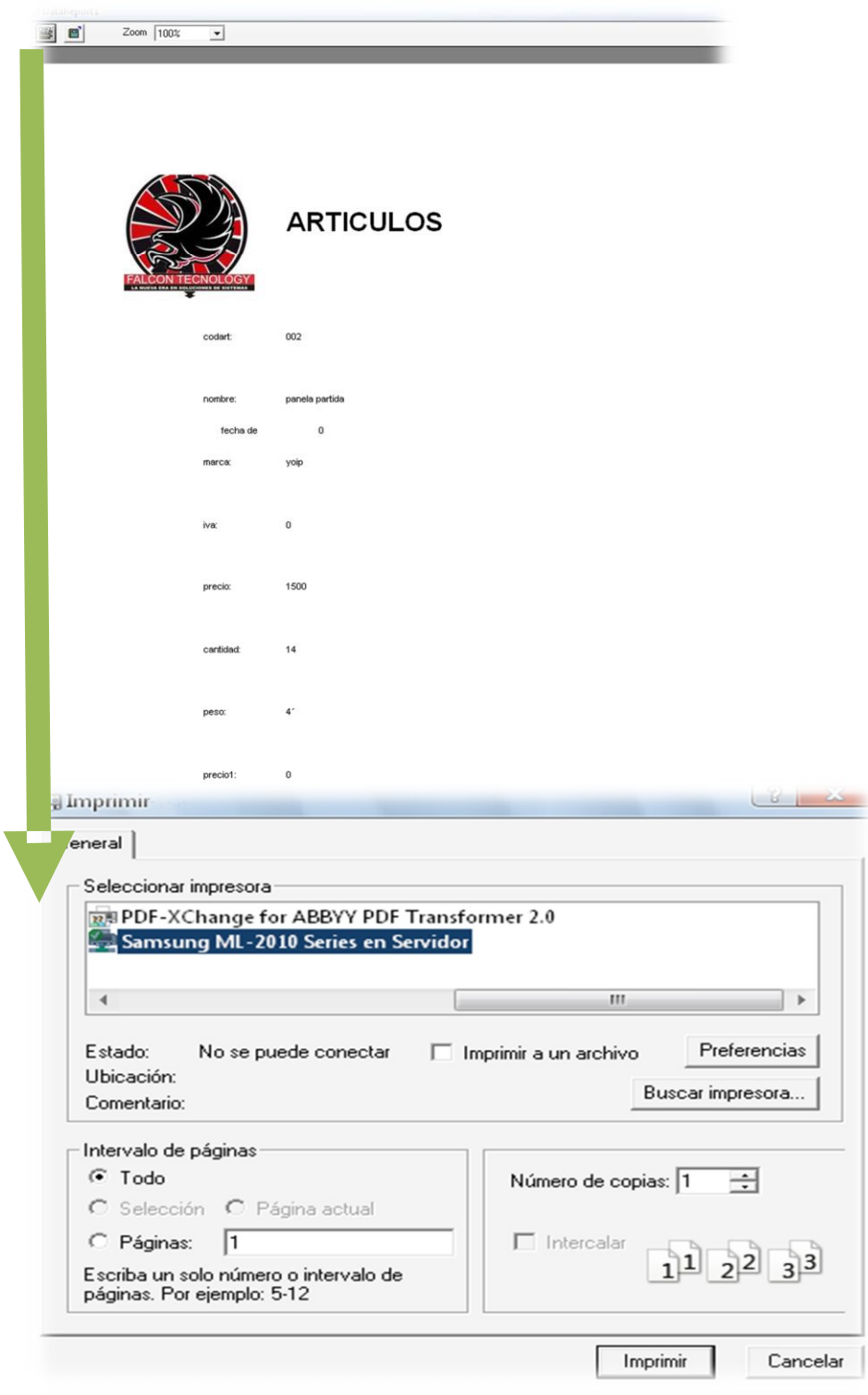

Luego la programación quedara así

Luego al darle click en el botón carga el reporte ya listo y programado para guardar e imprimir, si los datos no actualizan hay que programar lo siguiente

**Datareport1.Show**
**Dataenvironment.rsNombredelComando.Close**

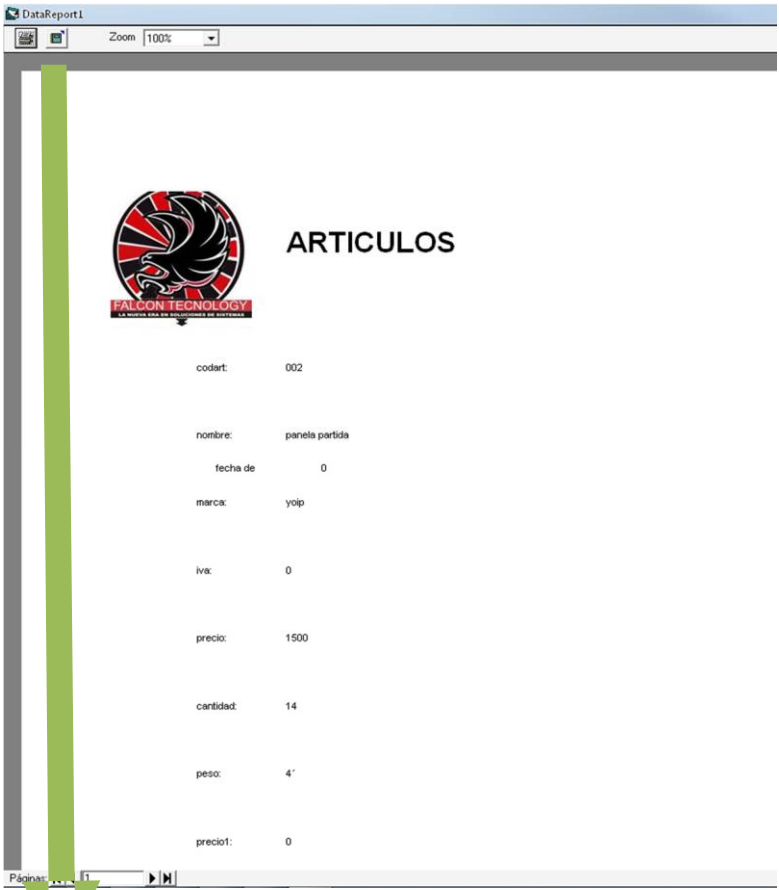


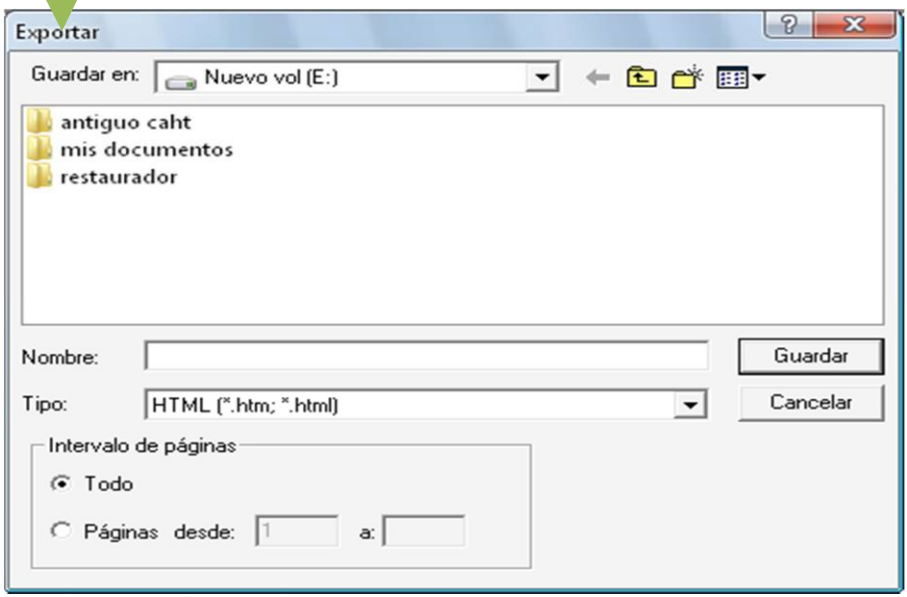

El reporte se verá así:




Imprimir

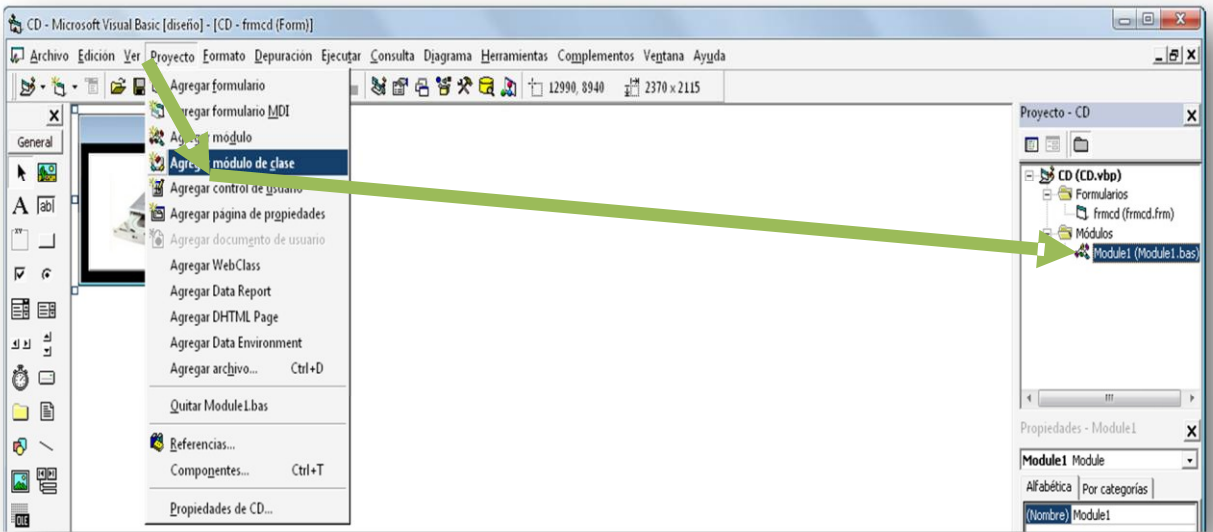
Guardar


DataReport1
Zoom 100%
FALCON TECNOLOGY
ARTICULOS
codart: 002
nombre: panela partida
fecha de 0
marca: yoip
iva: 0
precio: 1500
cantidad: 14
peso: 4´
precio1: 0


Exportar
Guardar en: Nuevo vol (E:)
antiguo caht
mis documentos
restaurador
Nombre:
Guardar
Tipo: HTML (*.htm; *.html)
Cancelar
Intervalo de páginas
Todo
Páginas desde: 1 a:

# 40. ABRIR Y CERRAR UNIDAD DE CD DESDE UN BOTON

Primero cargamos un modulo de windows de la siguiente manera click en el menú proyecto agregar módulo de clase luego; en ese módulo programo lo siguiente




```
Public Declare Function mciSendString Lib "winmm.dll" Alias "mciSendStringA" (ByVal
lpstrCommand As String, ByVal lpstrReturnString As String, _
ByVal uReturnLength As Long, ByVal hwndCallback As Long) As Long
```

Luego en el botón

La programación es esta

Form | Load

```
Private Sub Command1_Click()
On Error Resume Next
Command2.Visible = True
Command1.Visible = False
frmcd.Caption = "Abriendo. . . ."
retvalue = mciSendString("set CDAudio door open", returnstring, 127, 0)
End Sub
Private Sub Command2_Click()
On Error Resume Next
Command1.Visible = True
Command2.Visible = False
frmcd.Caption = "Cerrando. . . ."
retvalue = mciSendString("set CDAudio door closed", returnstring, 127, 0)
End Sub

Private Sub Form_Load()
On Error Resume Next
Command1.Visible = True
Command2.Visible = False
frmcd.Caption = "C D . . . "
End Sub
```

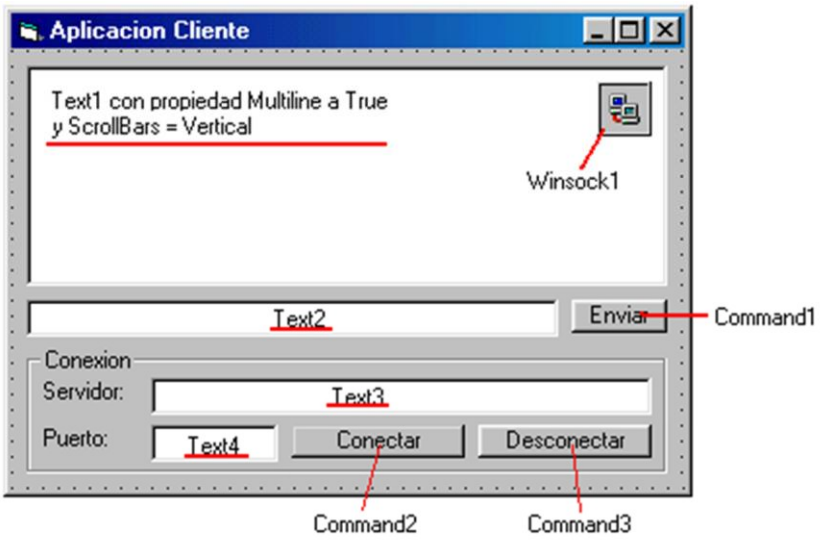
# 41. APLICACIONES CLIENTE SERVIDOR

Cómo programar aplicaciones cliente/servidor en visual Basic 6.0 utilizando el control winsock

Esta aplicación trabajara como un cliente simple que conecte a cualquier servidor, permita enviar texto plano y a la vez mostrar la información devuelta por este. Parecido a cómo trabajan los clientes de telnet.

### 1. Creando la interfaz del usuario

Realiza un formulario como el mostrado abajo, con los nombres por defecto de cada control y guarda el proyecto con el nombre "cliente.vbp".




### 2. Implementando la conexión

La primera acción a realizar y fundamental para toda aplicación de este tipo, es crear la conexión al servidor, ya que solo se puede transmitir información si la conexión cliente/servidor se encuentra activa.

Propiedades necesarias

- remotehost: asignamos la dirección a la que deseamos conectar.

- remoteport: asignamos el puerto al que deseamos conectar en remotehost.

Métodos necesarios

- connect(): conecta al servidor.

- close(): cierra la conexión al servidor.

Eventos involucrados

- connect(): ocurre cuando hemos establecido con éxito la conexión al servidor

- close(): ocurre cuando el servidor nos cierra la conexión.

- error(): ocurre en caso de errores.

Para realizar la conexión utilizamos el siguiente código:

```
Private sub command2_click()
'Asignamos los datos de conexion
Winsock1.remotehost = text3.text
Winsock1.remoteport = text4.text
'Conectamos el socket
Winsock1.close
Winsock1.connect
End sub
```

En las primeras dos líneas asignamos los datos de conexión al host remoto, como son la ip/dns (remotehost) y puerto (remoteport).

En la última línea llamamos al método "connect" para realizar la conexión, siempre asegurándonos que el socket no esté utilizándose. Para ello llamamos al método "close" que se encarga de cerrar toda conexión pendiente en el socket.

*Nota: también se puede especificar los datos de conexión (ip y puerto) directamente en el comando "connect" como parámetros, de la sig. Forma: winsock1.connect(host, puerto).*

Si la conexión se realiza con éxito se dispara un evento para tal fin, en donde podemos realizar acciones inmediatas en el momento preciso en que se logra establecer la conexión con el servidor. El evento es el siguiente:

```
Private sub winsock1_connect()
'Desplegamos un mensaje en la ventana
Text1.text = text1.text & _
"*** conexion establecida." & vbcrlf
'Desplazamos el scroll
Text1.selstart = len(text1.text)
End sub
```

En este momento ya tenemos creado los lazos básicos para realizar cualquier intercambio de datos con el servidor, ya sea texto ascii o datos binarios.

También hay que tener presente que en cualquier momento el servidor nos puede cerrar la conexión, o bien cerrarse por algún error, para ello es que contamos con el evento "close", que se dispara al perder la conexión con el servidor:

```
Private sub winsock1_close()
'Cierra la conexion
Winsock1.close
'Desplegamos un mensaje en la ventana
Text1.selstart = len(text1.text)
Text1.text = text1.text & "*** conexion cerrada por el servidor." & vbcrlf
Text1.selstart = len(text1.text)
End sub
```

Aquí solo desplegamos un mensaje en la pantalla informando del evento ocurrido, y cerrando previamente el socket para asegurarnos de que este actualice sus valores según el estado actual.

En cambio si queremos cerrar nosotros mismos la conexión con el servidor basta con llamar al método "close" directamente:

```
Private sub command3_click()

'Cierra la conexion

Winsock1.close

'Desplegamos un mensaje en la ventana

Text1.text = text1.text & _

"*** conexion cerrada por el usuario." & vbcrlf

'Desplazamos el scroll

Text1.selstart = len(text1.text)

End sub
```

**3. Enviando/recibiendo datos**

Una vez realizada con éxito nuestra conexión, solo resta comenzar a transferir datos, cabe mencionar que estos datos se envían siempre en forma binaria aunque sea solo texto, ya que el texto en si es una representación grafica de un numero binario, con esto quiero expresar que a través de un socket puedes enviar texto normal o datos binario, todos como variables de tipo string (cadenas).

Métodos necesarios

- senddata: envía datos al otro extremo de la conexión (socket remoto).

- getdata: recibe datos enviados por el extremo remoto (socket remoto).

Eventos involucrados

- dataarrival(): ocurre cuando el socket remoto nos está enviando datos.

- error(): ocurre en caso de errores.

Para enviar datos utilizamos el método "senddata" de la sig. Forma:

```
Private sub command1_click()
'Enviamos el contenido de text2
Winsock1.senddata text2.text & vbcrlf
'Apuntamos al final del contenido del textbox e
'Insertamos los nuevos datos obtenidos
Text1.selstart = len(text1.text) 'coloca el cursor al final del contenido
Text1.text = text1.text & "cliente >" & text2.text & vbcrlf 'mostramos los datos
Text1.selstart = len(text1.text) 'coloca el cursor al final del contenido
'Borramos text2
Text2.text = ""
End sub
```

Al método senddata solo se le pasa como parámetro el dato a enviar (en este caso el contenido de un textbox + los caracteres de nueva línea y retorno de carro) y este lo envía inmediatamente al socket remoto.

Cuando el socket remoto nos envía un dato (de la misma forma que realizamos anteriormente) se nos genera el evento "dataarrival()" indicando que tenemos nueva información disponible, y esta información la cogemos con el método "getdata":

```
Private sub winsock1_dataarrival(byval bytestotal as long)
Dim buffer as string 'variable para guardar los datos
'Obtenemos los datos y los guardamos en una variable
Winsock1.getdata buffer
'Apuntamos al final del contenido del textbox e
'Insertamos los nuevos datos obtenidos
Text1.selstart = len(text1.text) 'coloca el cursor al final del contenido
Text1.text = text1.text & "servidor >" & buffer 'mostramos los datos
Text1.selstart = len(text1.text) 'coloca el cursor al final del contenido
End sub
```

En este ejemplo solo obtenemos los datos y lo mostramos inmediatamente en la ventana del cliente, no hacemos ningún tratamiento previo de los datos, como sería lo habitual.

**4. Manejo de errores**

Es muy importante tomar alguna acción cuando se produzca algún error, aunque esta acción tan solo sea cerrar la conexión e informar al usuario de lo ocurrido.

Para el manejo de errores producidos durante la conexión contamos con un evento dedicado, llamado "error()" el cual retorna varios valores para darnos información al respecto, entre ellos los más comunes son:

| | |
|---|---|
| Number as integer | Informa sobre el numero del error producido |
| Description as string | Entrega una breve descripción del error |

En caso de producirse algún error la acción más simple de realizar es simplemente cerrar la conexión con el método "close":

```
Private sub winsock1_error(byval number as integer, description as string, byval scode as long, byval source as string, byval helpfile as string, byval helpcontext as long, canceldisplay as boolean)

'Cerramos la conexion

Winsock1.close

'Mostramos informacion sobre el error

Msgbox "error numero " & number & ": " & description, vbcritical

End sub
```

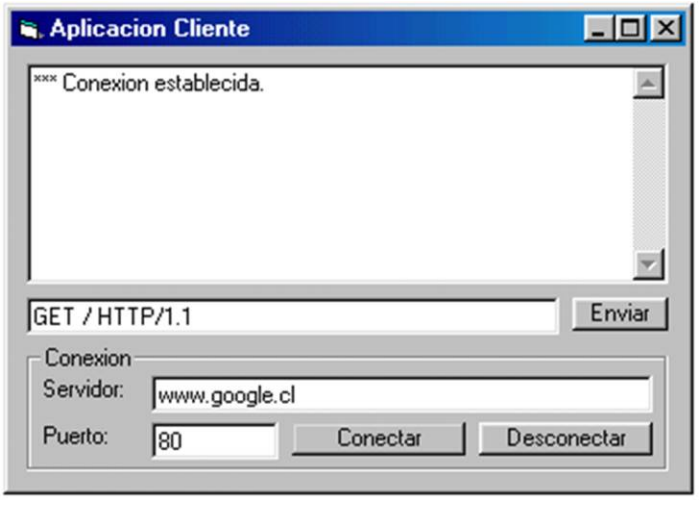
**5. Prueba de la aplicación**

En este punto ya estamos listo para comenzar a usar nuestro programa cliente, solo le damos a ejecutar desde el entorno del visual basic o bien compilamos y ejecutamos el archivo.

Para asegurarnos que todo ha salido bien, vamos a realizar una pequeña prueba, conectaremos a www.google.com y solicitaremos la página de inicio:

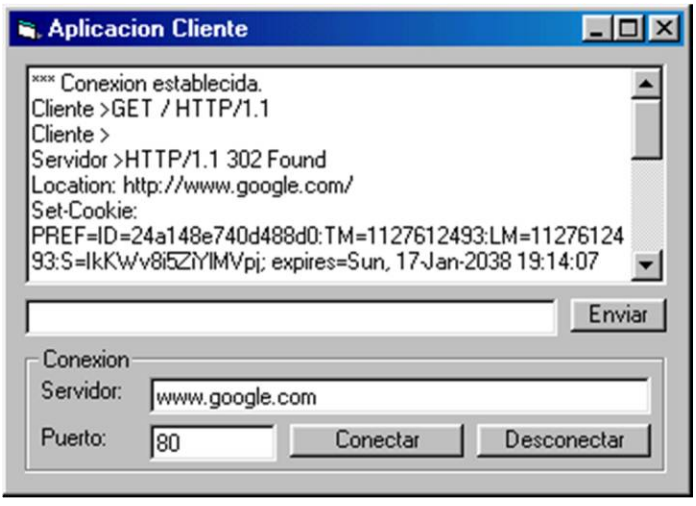
En el campo "servidor" de nuestro programa escribimos "www.google.com", y en el campo "puerto" colocamos el "80". Le damos al botón "conectar".

Si todo va bien, deberíamos obtener el mensaje "conexión establecida", si es así entonces en el campo de enviar texto, escribimos "get / http/1.1":




Y para enviar presionamos dos veces el botón "enviar". La razón de esto es para que envié la cadena que escribimos más dos caracteres de retorno de carro o nueva línea (vbcrlf), esto por especificaciones del protocolo http (que es lo que estamos utilizando aquí).

Deberíamos ver algo como esto:




Si recibimos texto desde el servidor (las cadenas que inician con "servidor >") es que nuestro cliente funciona perfectamente y hemos realizado la conexión, enviado y recibido datos con éxito. Ya podemos descasar un rato y celebrar :).

**Mi primera aplicación servidor**

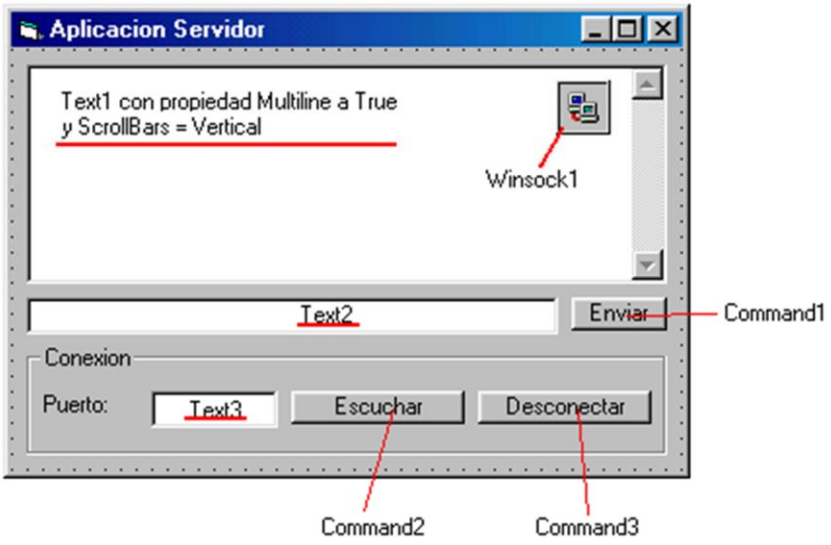
Vamos a realizar una aplicación que se mantenga a la escucha de una conexión entrante y la acepte, podrá enviar y recibir datos desde el cliente.

Al principio será mono-conexión, es decir, solo permitirá una conexión a la vez al servidor, pero luego la implementaremos para múltiples conexiones.

Algunas partes del código para el servidor son idénticas al del cliente (realizado anteriormente) así que solo me limitare a mostrar cómo queda el código y la explicación de la misma se entenderá que es la correspondiente a la del cliente.

**1. Creando la interfaz del usuario**

Realiza un formulario como el mostrado abajo, con los nombres por defecto de cada control y guarda el proyecto con el nombre "servidor.vbp".




## 2. Implementando la conexión

Al igual que en el cliente, lo primero es habilitar el socket para que pueda quedar esperando una conexión, se dice que queda "a la escucha de". Para esto solo necesitamos un botón "escuchar" y como datos un puerto local (a elección) en el cual deseamos recibir conexiones entrantes.

Propiedades necesarias

- localport: asignamos el puerto local en el cual deseamos recibir conexiones.

Métodos necesarios

- listen(): escucha peticiones entrantes.

- close(): cierra la conexión al servidor.

Eventos involucrados

- connectionrequest(): ocurre cuando un cliente nos solicita una conexión al servidor.

- close(): ocurre cuando el servidor nos cierra la conexión.

- error(): ocurre en caso de errores.

El código utilizado para el botón "escuchar" es el siguiente:

```
Private sub command2_click()
'Cerramos cualquier conexion previa
Winsock1.close
'Asignamos el puerto local que abriremos
Winsock1.localport = text3.text
'Deja el socket esuchando conexiones
Winsock1.listen
'Desplegamos un mensaje en la ventana
Text1.selstart = len(text1.text)
Text1.text = text1.text & "*** esuchando conexiones." & vbcrlf
Text1.selstart = len(text1.text)
End sub
```

La primera línea de código cierra la conexión actual, para luego poder modificar los datos y crear una nueva conexión sin que nos de errores.

La siguiente línea le dice en que puerto deseamos recibir conexiones, y luego llama al socket para que quede a la escucha de conexiones en ese puerto.

Hasta aquí el socket solo está "escuchando" conexiones, es decir aun nadie se puede conectar al servidor completamente porque no se ha implementado esa parte por el momento.

Esto solo nos permite avisarnos cada vez que un cliente se quiera conectar o bien cada vez que un cliente "solicita una conexión entrante". Cuando este sucede se genera el evento "connectionrequest()":

```
Private sub winsock1_connectionrequest(byval requestid as long)
'Mostramos un mensaje en la ventana
Text1.selstart = len(text1.text)
Text1.text = text1.text & "*** peticion numero " & requestid & vbcrlf
Text1.selstart = len(text1.text)
'Cerramos previamente el socket
Winsock1.close
'Aceptamos la conexion
Winsock1.accept requestid
'Desplegamos un mensaje en la ventana
Text1.selstart = len(text1.text)
```

```
Text1.text = text1.text & "*** conexion aceptada, listo para interactuar." & vbcrlf

Text1.selstart = len(text1.text)

End sub
```

Con estas líneas ya estamos conectado completamente, pero de seguro que quedan muchas dudas, y precisamente este punto es uno de los más importantes, así que pasare a detallar cada cosa.

Lo primero que habíamos realizado es dejar el socket a la escucha de conexiones (para esto utilizamos el método "listen"). Con esto ya tenemos un puerto abierto y atento a toda actividad.

Cuando un "cliente" se intenta conectar a ese puerto, el socket lo detectara y para ello generara el evento "connectionrequest()" que significa "petición de conexión" y además le asigna una identidad a esa "petición" que identifica al "cliente" remoto. Esta identidad es pasada como parámetro en el evento "connectionrequest()" con el nombre de "requestid" y es de tipo "long".

Cuando se genera el evento lo que tenemos que hacer es "aceptar" la conexión entrante "requestid" mediante el metodo "accept", si no lo hacemos al llegar al "end sub" del evento, la conexión del "cliente" será cerrada automáticamente.

Algo interesante es ver que antes de aceptar la conexión con "accept" primero cerramos la conexión con "close", esto que puede parecer ilógico no lo es, porque el socket lo teníamos ocupado y activo "escuchando conexiones", y ahora necesitamos que establezca una conexión par a par con el cliente, por ello es que cerramos la función de "escuchar conexiones del socket" y le decimos que

acepte la conexión entrante y así automáticamente se conecta en forma directa con el cliente y ya no entenderá nuevas conexiones entrantes. (no puede realizar dos funciones a la vez)

Para cerrar la conexión basta con usar el método "close" en cualquier momento:

```
Private sub command3_click()
'Cierra la conexion
Winsock1.close
'Desplegamos un mensaje en la ventana
Text1.selstart = len(text1.text)
Text1.text = text1.text & "*** conexion cerrada por el usuario." & vbcrlf
Text1.selstart = len(text1.text)
End sub
```

```
Private sub winsock1_close()
'Cierra la conexion
Winsock1.close
'Desplegamos un mensaje en la ventana
Text1.selstart = len(text1.text)
```

```
Text1.text = text1.text & "*** conexion cerrada por el cliente." & vbcrlf

Text1.selstart = len(text1.text)

End sub
```

**3. Enviando/recibiendo datos**

Esto es idéntico al explicado en la parte del cliente:

```
Private sub command1_click()

'Enviamos el contenido de text2

Winsock1.senddata text2.text & vbcrlf

'Apuntamos al final del contenido del textbox e

'Insertamos los nuevos datos obtenidos

Text1.selstart = len(text1.text) 'coloca el cursor al final del contenido

Text1.text = text1.text & "servidor >" & text2.text & vbcrlf 'mostramos los datos

Text1.selstart = len(text1.text) 'coloca el cursor al final del contenido

'Borramos text2

Text2.text = ""

End sub
```

```
Private sub winsock1_dataarrival(byval bytestotal as long)

Dim buffer as string 'variable para guardar los datos

'Obtenemos los datos y los guardamos en una variable

Winsock1.getdata buffer

'Apuntamos al final del contenido del textbox e

'Insertamos los nuevos datos obtenidos

Text1.selstart = len(text1.text) 'coloca el cursor al final del contenido

Text1.text = text1.text & "cliente >" & buffer 'mostramos los datos

Text1.selstart = len(text1.text) 'coloca el cursor al final del contenido

End sub
```

**4. Manejo de errores**

Esto es idéntico al explicado en la parte del cliente:

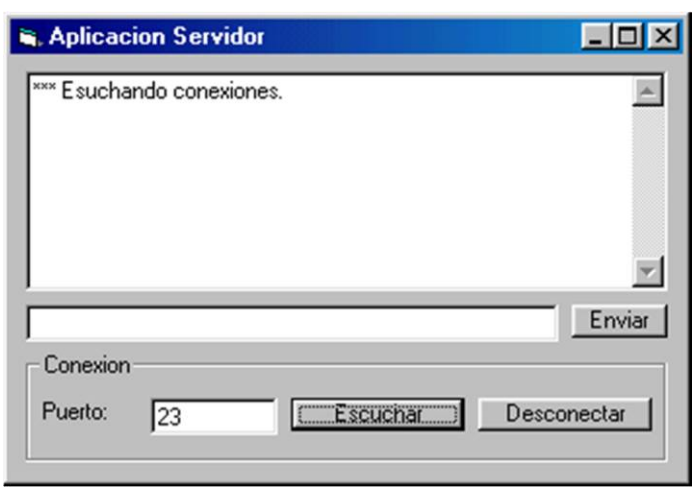
```
Private sub winsock1_error(byval number as integer, description as string, byval scode as long, byval source as string, byval helpfile as string, byval helpcontext as long, canceldisplay as boolean)

'Cerramos la conexion

Winsock1.close
```

```
'Mostramos informacion sobre el error

Msgbox "error numero " & number & ": " & description, vbcritical

End sub
```

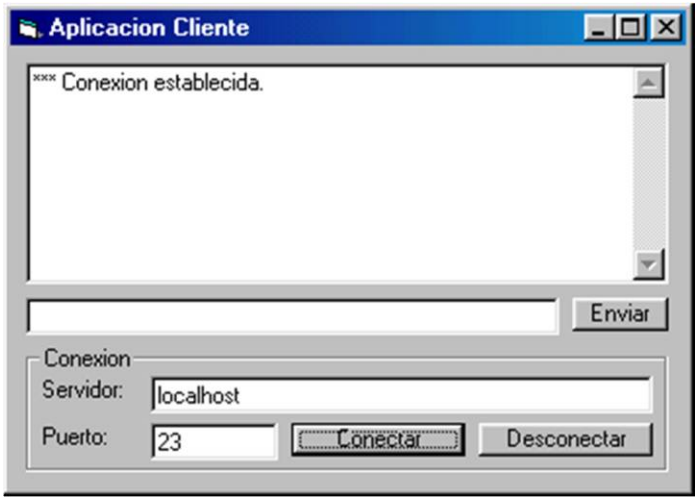
**5. Prueba de la aplicación**

Después de mucho "copiar/pegar" :), ya estamos listos con nuestra aplicación servidor y listos para realizar las primeras pruebas y ver si todo trabajacorrectamente.

Lo primero, ejecuta la aplicación servidor y donde dice "puerto" coloca cualquier número de los 65.535 disponibles, por ej. El "23". Luego dale al botón "escuchar":

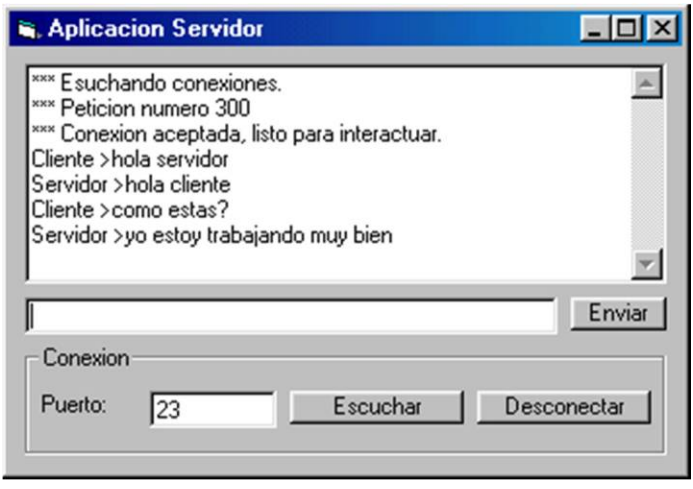


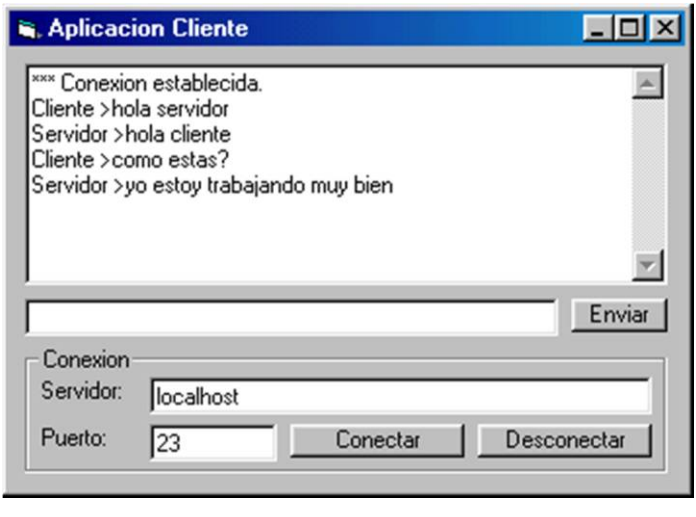

Ahora ejecuta la aplicación cliente y en "servidor" coloca "localhost" o "127.0.0.1" y en "puerto" coloca el "23". Dale al botón "conectar".



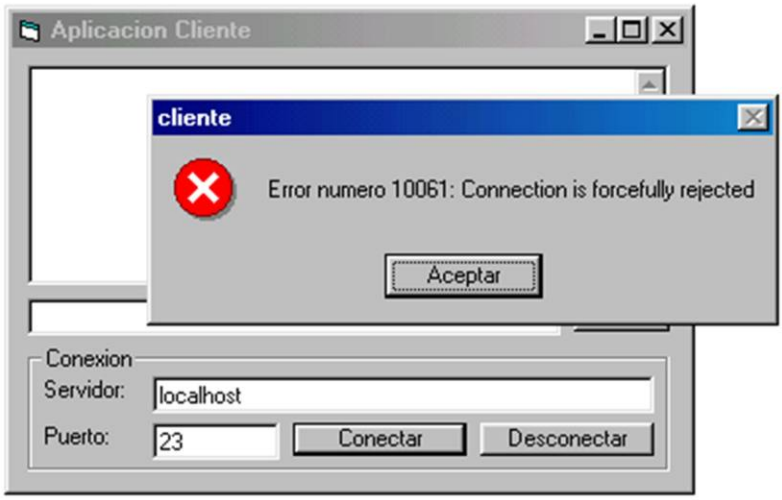

En el servidor obtienes:

Ya estamos listos y trabajando con nuestra aplicación cliente/servidor!!, ¿no lo crees?, prueba a enviar texto entre cliente->servidor y servidor->cliente y compruébalo tú mismo:




Ahora que sabemos que todo trabaja correctamente te invito a hacer una prueba más. Con la conexión establecida y funcionando de par a par ente cliente/servidor, ejecuta una nueva aplicación "cliente" e intenta conectar al servidor en el mismo puerto (en este caso servidor "localhost" y puerto "23"), y espera los resultados:




Nos dice que no logra establecer la conexión, este es el mismo mensaje que entrega si el servidor al que intenta conectar no tiene ningún puerto abierto!!, lo que sucede es que el servidor ya no se encuentra "a la escucha de conexiones" y por lo tanto no atenderá nuevas peticiones de conexión.

**Aplicación servidor multi-conexión**

Ahora nos encontramos con los conocimientos suficientes para implementar un servidor que pueda aceptar un número indefinido de conexiones entrantes.

En este proyecto usaremos el mismo código fuente del servidor mono-conexión utilizado anteriormente, ya que los cambios son muy pocos en realidad.

Bien, entonces abrimos el proyecto del servidor mono-conexión y lo guardamos como "servidor multi.vbp" para comenzar a trabajar.

La interfaz la dejaremos tal cual, todo el cambio será en relación al código mismo y a la modificación de algunos controles. También trabajaremos con arreglo de controles, si nunca lo has hecho podrías buscar un poco de información al respecto o bien intentar seguir adelante, que lo explicare de forma breve.

## 1. Vista general del funcionamiento

Como vimos anteriormente en el servidor mono-conexión, dejábamos un socket a la escucha de conexiones entrantes, y al recibir una petición de conexión (evento "connectionrequest") le decíamos al winsock que aceptara esa identidad y este a su vez establecía una conexión con el cliente.

Los principios para crear un servidor multi-conexión son los mismos, salvo que necesitamos dejar un socket escuchando permanentemente conexiones entrantes, este nunca se debe cerrar (al contrario de lo que pasaba en el caso del servidor mono-conexión), entonces ¿como podemos aceptar una conexión si no podemos cerrar el socket que tenemos a la escucha?, ¿acaso podemos dejar un mismo socket escuchando conexiones y atendiendo otra a la vez?, la respuesta es no podemos, pero nada nos impide hacer que otro socket que se encuentra inactivo acepte y atienda una petición de conexión. De esta forma el trabajo total se reparte entre varios sockets: un socket permanentemente escuchando peticiones de conexión (recepcionista) y otros tantos socket que se encargan de atender a cada uno de los clientes (ejecutivos).

## 2. Creando el arreglo de winsocks

Para poder trabajar con varias conexiones a la vez necesitamos varios sockets disponibles, ya que cada uno solo puede trabajar con una sola conexión, y como en principio no conocemos la cantidad de winsocks que necesitaremos debemos inclinarnos por crear arreglos de controles winsock e irlos cargando dinámicamente.

Si lo deseas también puedes crear una n cantidad de winsocks y solo trabajar con ellos, pero tu número máximo de conexiones posibles será el máximo de winsocks que tengas.

Entonces, para crear el arreglo debemos seguir los sig. Pasos:

1. Agregar un nuevo winsock al formulario (winsock2)
2. Copiar el control (clic derecho sobre este y seleccionar "copiar")
3. Pegar el control en el formulario, y cuando pregunte por si deseas crear el arreglo dile que "si". (clic derecho sobre el formulario y seleccionar "pegar")
4. Borramos el nuevo winsock que se ha creado (winsock2(1)).

La razón de borrar este último winsock es porque no nos hace falta, como mencionamos en un principio, nosotros crearemos los winsocks necesarios de forma dinámica, solo necesitamos tener existente el winsock2 de índice cero.

**3. Limpiando código innecesario**

En realidad no hay código innecesario pero si código que debe cambiar de lugar, como veremos en su momento, por ahora solo nos limitaremos a borrar todo el código del evento "winsock1_dataarrival" ya que nunca lo usaremos en este winsock porque solo trabajara como repartidor de trabajoy "winsock1_connectionrequest" que será implementada mas adelante, lo mismo para la acción del botón "enviar" (command1_click).

**4. Enviando/recibiendo datos**

Vamos a ver como se realizan las acciones de recibir y enviar datos cuando tenemos arreglos de sockets (winsock2()), nos guardaremos laadministración de las peticiones de conexiones para más adelante.

Para enviar datos (mediante el botón "enviar") podemos hacerlo directamente con algún socket específico, definiendo su identidad, o bien con todos los sockets recorriendo el arreglo. Esto último es lo que haremos a modo de ejemplo:

```
Private sub command1_click()
Dim numelementos as integer 'numero de sockets
Dim i as integer 'contador
'Obtiene la cantidad de winsocks que tenemos
Numelementos = winsock2.ubound
'Recorre el arreglo de sockets
For i = 0 to numelementos
'Si el socket se encuentra conectado...
If winsock2(i).state = sckconnected then
'Enviamos el contenido de text2
Winsock2(i).senddata text2.text & vbcrlf
'Apuntamos al final del contenido del textbox e
'Insertamos los nuevos datos obtenidos
```

```
Text1.selstart = len(text1.text) 'coloca el cursor al final del contenido

Text1.text = text1.text & "sock" & i & ":servidor >" & text2.text & vbcrlf 'mostramos los datos

Text1.selstart = len(text1.text) 'coloca el cursor al final del contenido

End if

Next

'Borramos text2

Text2.text = ""

End sub
```

Y cuando recibamos datos desde el cliente se nos generara el evento "winsock2_dataarrival" que ahora incluye un nuevo parámetro "index" de tipo "integer" y contiene el número o índice del socket que genera el evento (todo en relación al arreglo de sockets).

Para recibir datos solo tenemos que tomar en cuenta ese "index" y el resto es igual a lo visto en el servidor de conexión mono-usuario:

```
Private sub winsock2_dataarrival(index as integer, byval bytestotal as long)

Dim buffer as string 'variable para guardar los datos

'Obtenemos los datos y los guardamos en una variable
```

```
Winsock2(index).getdata buffer

'Apuntamos al final del contenido del textbox e

'Insertamos los nuevos datos obtenidos

Text1.selstart = len(text1.text) 'coloca el cursor al final del contenido

Text1.text = text1.text & "sock" & index & ":cliente >" & buffer 'mostramos los datos

Text1.selstart = len(text1.text) 'coloca el cursor al final del contenido

End sub
```

**5. Evento error y close**

El código para los eventos "error" y "close" del arreglo de sockets es muy simple:

```
Private sub winsock2_close(index as integer)

'Cierra la conexion

Winsock2(index).close

'Desplegamos un mensaje en la ventana

Text1.selstart = len(text1.text)

Text1.text = text1.text & "sock" & index & ":*** conexion cerrada por el cliente." & vbcrlf

Text1.selstart = len(text1.text)
```

```
End sub
```

```
Private sub winsock2_error(index as integer, byval number as integer, description as string, byval scode as long, byval source as string, byval helpfile as string, byval helpcontext as long, canceldisplay as boolean)

'Cerramos la conexion

Winsock2(index).close

'Mostramos informacion sobre el error

Msgbox "error numero " & number & ": " & description, vbcritical

End sub
```

**5. Escuchando y atendiendo a las conexiones**

El siguiente paso será recibir las peticiones de conexión y asignárselas a cada socket para que las atienda. Para ello necesitaremos crear un nuevo socket cada vez que recibamos una petición de conexión y decirle que acepte la identidad de la conexión.

Para facilitar las cosas nosotros haremos una función que se encargue de crear los sockets y que además devuelva el número del nuevo socket creado:

```
'Carga un nuevo socket al arreglo y devuelve su indice

Private function nuevosocket() as integer
```

```
Dim numelementos as integer 'numero de sockets

Dim i as integer 'contador

'Obtiene la cantidad de winsocks que tenemos

Numelementos = winsock2.ubound

'Recorre el arreglo de sockets

For i = 0 to numelementos

'Si algun socket ya creado está inactivo

'Utiliza este mismo para la nueva conexion

If winsock2(i).state = sckclosed then

Nuevosocket = i 'retorna el indice

Exit function 'abandona la funcion

End if

Next

'Si no encuentra sockets inactivos

'Crea uno nuevo y devuelve su identidad

Load winsock2(numelementos + 1) 'carga un nuevo socket al arreglo

'Devuelve el nuevo indice

Nuevosocket = winsock2.ubound
```

```
End function
```

Esta función no solo crea un nuevo socket, sino que además si encuentra alguno que se había creado antes y este se encuentra inactivo (desconectado) lo selecciona para volverlo a utilizar y así aprovechar más los recursos del sistema.

*Nota: esta no es la forma más óptima de manejar arreglos de objetos, ya que no nos permite ir liberando de la memoria (borrando) los sockets que ya no son utilizados y solo se limita a crear nuevos.*

Ahora nos situamos en el evento "connectionrequest()" del "winsock1" (el que hará de recepción) y escribimos el siguiente código:

```
Private sub winsock1_connectionrequest(byval requestid as long)

Dim numsocket as integer 'el numero del socket

'Mostramos un mensaje en la ventana

Text1.selstart = len(text1.text)

Text1.text = text1.text & "*** peticion numero " & requestid & vbcrlf

Text1.selstart = len(text1.text)

'Creamos un nuevo socket

Numsocket = nuevosocket
```

```
'Aceptamos la conexion con el nuevo socket

Winsock2(numsocket).accept requestid

'Desplegamos un mensaje en la ventana

Text1.selstart = len(text1.text)

Text1.text = text1.text & "sock" & numsocket & ":*** conexion aceptada, listo Para interactuar." & vbcrlf

Text1.selstart = len(text1.text)

End sub
```

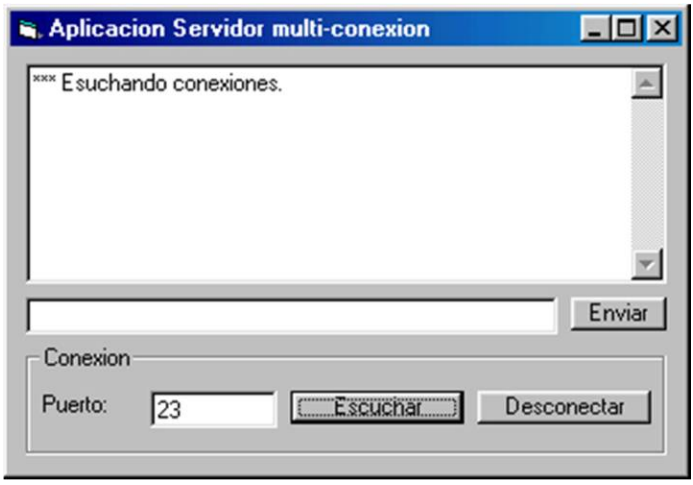
Aquí lo primero es crear un nuevo socket (o reutilizar alguno disponible) y decirle a ese socket que acepte aquella conexión. Una vez realizado esto ya estamos libres nuevamente para recibir otra conexión.

También es posible denegar conexiones realizadas desde alguna ip especifica, para ello solo hay que revisar la propiedad "winsock1.remotehostip" y ver si aceptamos su conexión o bien se la rechazamos con "close".

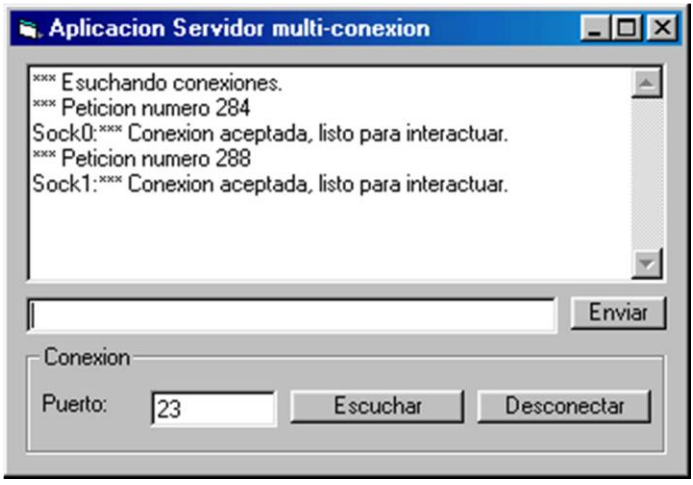
**6. Prueba de la aplicación**

Ahora que todo parece estar completo, realizaremos la prueba del servidor.

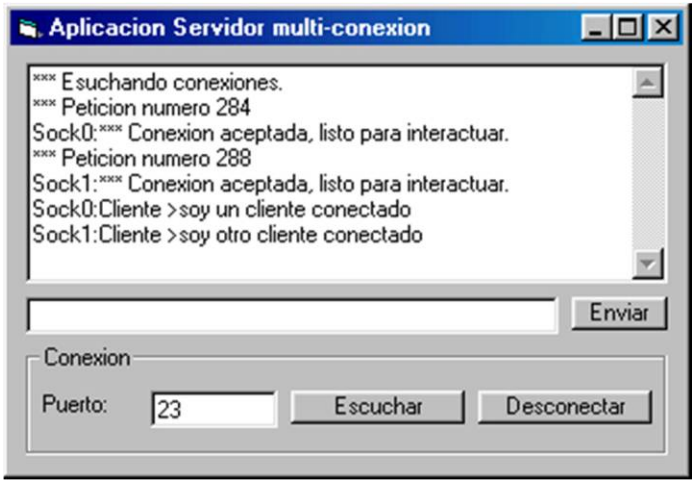
Ejecuta la aplicación, como "puerto" coloca el "23" y dale al botón "escuchar".




Ahora ejecuta dos aplicaciones "cliente" y conéctalos al puerto "23" de "localhost". Notaras que ya no da el error que vimos con el servidor mono-conexión y que ambos se encuentran conectados:




Prueba a enviar mensajes entre ellos y veras que todo trabaja perfectamente!!, puedes identificar cada conexión porque en el mensaje aparece "sockn" donde "n" es el índice del socket, así sabrás en cada momento que socket es el que está enviando el mensaje:




Y ya funciona todo ok!.

**Conclusión**

Hemos llegado al final de este tutorial, y hemos aprendido a realizar conexión cliente/servidor mono y multi-conexiones de forma básica, digo básica porque hay mejores maneras de implementarlas y más minuciosas, pero esta es la base de todas ellas, y el resto lo obtendrás por la práctica.

Espero que este texto les haya sido de utilidad y cualquier duda o errores encontrados no dudes en comunicármelo, que lo corregiré tan pronto como pueda.